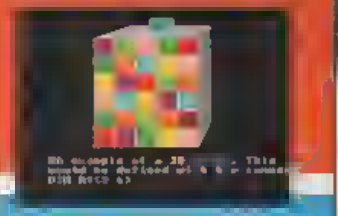
FREE!
WITH CU AMIGA

PART FIVE • MAY 1993



FIRST STEPS P10

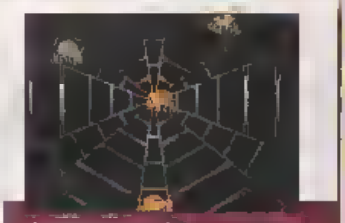
AMIGA Guide

THE COMPLETE GUIDE TO THE AMIGA

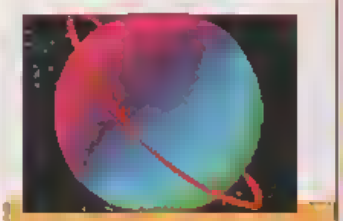
AMOS SPECIAL

THE DEFINITIVE GUIDE TO AMOS

TEXT HANDLING P12



SPRITES P18



SCREEN CONTROL P20

AMOS COMPILER P24

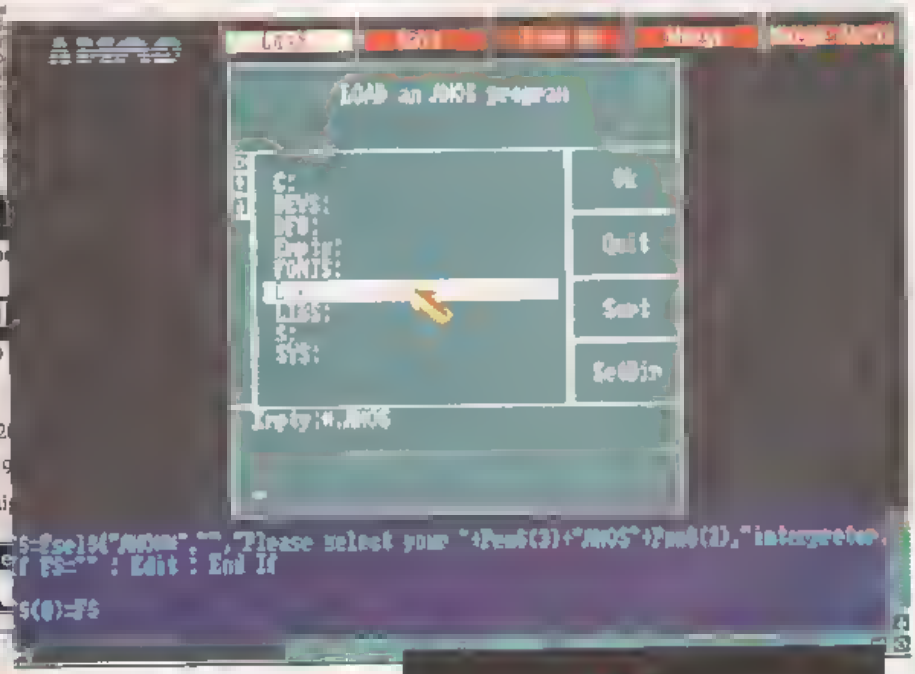
GET THE MOST OUT OF AMOS IN OUR STEP-BY-STEP GUIDE

32 PAGE SPECIAL

FREE! EXCLUSIVE TO CU AMIGA MAGAZINE! PART FIVE OF THE MOST COMPREHENSIVE GUIDE TO THE AMIGA EVER PUBLISHED.

CONTENTS

No doubt you've already been experimenting with last month's fabulous coverdisk giveaway, AMOS. In this supplement, we show you exactly what the program is capable of and how to get the best out it.



The AMOS system is very similar to most other Workbench-friendly utilities, so you should have no trouble at all getting to grips with it.

With a minimum of fuse and a little planning, you could be writing games as good as this. Honest!

Editorial

Welcome to the fifth in the series of CU Amiga supplements. As you have already noticed this one is dedicated to the superb Amiga programming utility AMOS, given away on last month's coverdisk (if you don't have it, contact the back issues department immediately on 0858 410510) and aims to show you how to get the most from it.

Everyone has wanted to write their own games and utilities at some point. Perhaps you remember playing *Le Mans* on your old Commodore 64 and would love to play it again, only can't find a version of it anywhere on PD. Or you might want an address book/database but can never find one to suit your needs. Until now the only options have been to wait for the right package to appear or get the next best thing. With AMOS, however, you'll be able to create the package you want, to your specifications and with all the extras and bonuses you need. Sounds too good to be true? Not at all.

One thing to bear in mind is that AMOS can't do everything. Because it's a high level programming language (see pages 4-5 for more information), programs written in AMOS will never run as fast as machine code programs. Being the thoughtful bunch that we are though, we were kind enough to include the AMOS compiler on last month's disk too, which helps speed up the running of your self-written software no end. Page 24 explains in more detail how the compiler works and what it can do for you.

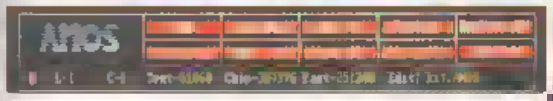
If you're a beginner to programming, or the Amiga in general, don't break into a sweat at the sight of all the listings and technical terms used in this guide. AMOS is the perfect tool for the beginner, giving you enough power to create some seriously impressive stuff, while at the same time keeping things simple enough for you to understand what you're doing.

In this guide, we'll take you through the basics of good game design, including a checklist of things to do before you start programming. We'll look at program construction and some of the data handling commands needed by every program. Once you've worked through those two, we can start doing some really interesting stuff, including AMOS's powerful graphic and sound commands.

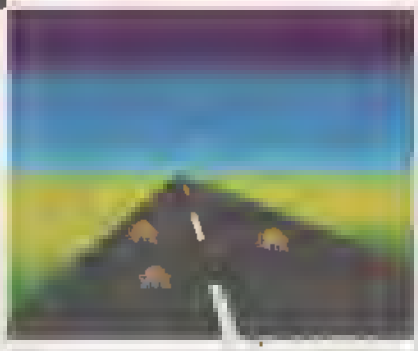
You'll learn how to add control routines to your programs, and how to achieve a whole host of unusual effects. And if all that isn't enough, we'll even show you how to turn your programs into standalone files, ready to be released onto the PD market or even onto a CU coverdisk if they're good enough.

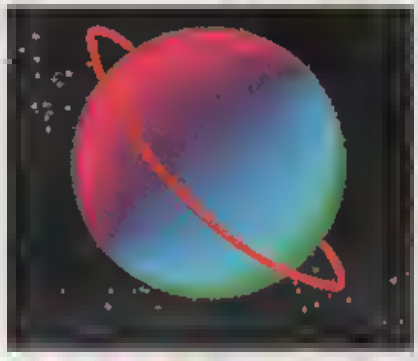
Programming is no longer purely the domain of the highly qualified expert. It's an exciting hobby that can often throw more challenges your way than *Monkey Island 2*. No doubt you can't wait to start, so let's go!

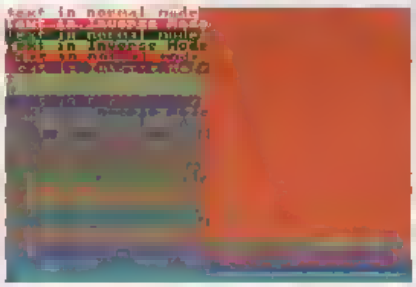
contents



All the important instructions can be found in the menu bars at the top of the screen. See page 6 for more details







EDITOR
Tony Dillon

COMPUTER GRAPHICS
Paul Fleal

WRITTEN BY
Tony Dillon

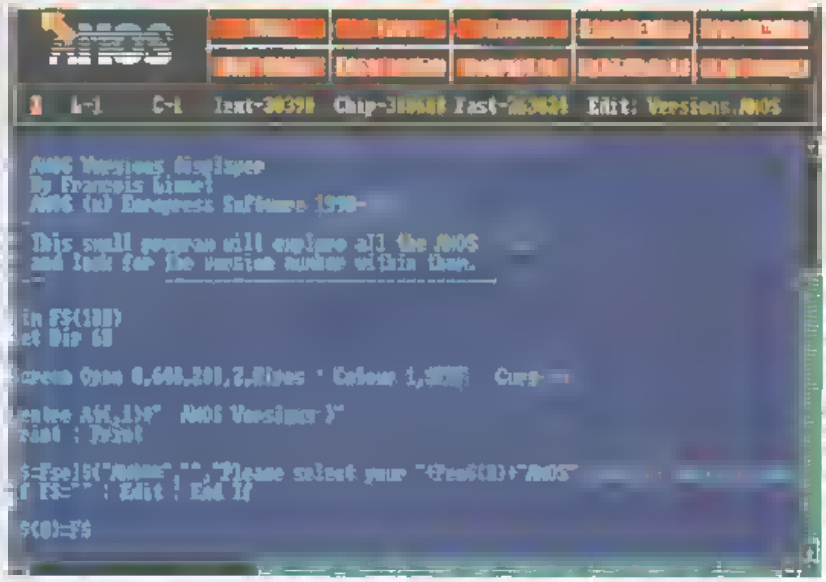
PUBLISHER
Garry Williams

AMIGA GUIDE
CU AMIGA
EMAP IMAGES
30-32 FARRINGDON LANE
LONDON EC1R 3AU

This issue of the Amiga Guide is free with the May 1993 issue of CU Amiga and must not be sold separately.

INTRODUCTION

Now that you've got AMOS loaded, you're probably wondering exactly what it is and what it's capable of. Here's the place to find out...



The AMOS editor screen. Get used to it, because you'll be seeing a lot of it in the future!

AMOS is intended first and foremost as a game creation package, but it can be used for far more. Stepping around the usual problems associated with coding (tedious routine writing masses of indecipherable codes and figures) AMOS uses an English-based parser to give novices the tool to create anything they want. A gateway to your Amiga's power, if you like.

In the past, game creators have generally been extremely limiting. You can spot a game written in *The Quill* and *3D Construction Kit* a mile off. Some say you can spot a game written in AMOS a mile off, but this is untrue. With AMOS, you can write anything you want, with only your imagination and experience shaping any barriers. To begin with, you probably won't be able to make *Midwinter 4*, or *Elite 3*, but don't let that stop you from trying.

PHRASE BOOK

AMOS itself is a compilation of two programming languages, the AMOS BASIC interpreter and the AMAL command language. The interpreter handles almost all of your code, turning your English commands into machine language for the processor to run. AMAL is the sprite animation suite that handles all the sprite and Bob (Blitter Object) routines. Together, they are capable of some fairly amazing things. But before you can use them, you need to understand how they work.

The interpreter is a derivative of BASIC (Beginners All purpose Symbolic Instruction Code), the age-old language favoured by both schools and novices. Rather than try and work the processor directly, the interpreter provides you with a suite of over 500 English commands, which make for readable listings and a good understanding of what things do right.

TABLE 1

```
PRINT "What is your name?"
INPUT Answer$
IF Answer$="Dan Slingsby" THEN PRINT "Hello Dan, Fancy meeting you here."
IF Answer$<>"Dan Slingsby" THEN PRINT "Have you seen Dan anywhere?"
```

THE FILE SELECTOR

The file selector can look a little daunting at first, but don't worry. It's easier than it looks.

It's probably worth your while getting to know the file selector, as it's something you'll be seeing a lot of. It works in much the same way as any other, with the slider bar on the left sliding through the list of files, and the standard 'OK', 'Sort' and 'Quit' buttons on the right. But how do you actually use it?

First, click on the 'Load' button on the main menu bar. The file selector appears, and after a moment the list of files appears. If the file that you want is listed, fine. Just double-click on its name to load it.

If, however, the file you want is in a directory, then you need to single-click on the directory name to open it, and then double-click on the name of the required file.

If your file is on a completely different disk then remove the disk in the drive and insert the disk with your file on. Click on the small button above the slider arrows with the right mouse button to get a list of devices, and then click on the name of the newly inserted disk.

INTRO

TO AMOS

from the start. The listing shown in Table 1 is a good example. You can probably already tell what that program will do when you run it.

THE EDIT SCREEN

Load up AMOS as shown last month (page 14), and take a good look at the main screen, known as the Edit Screen. This is where all the hard work happens. The strip at the top is the menu bar, and we'll be looking at that in just a moment. Below that is the information line, which tells you various things about your system at a glance.

I (Or O): Whether the editor is in Insert or Overwrite mode
L=1: Current line
C=1: Current column
Text=: The amount of memory assigned to the editor
Chip=: The amount of chip memory free
Fast=: The amount of fast memory free
Edit=: The name of the current program

Along the side and bottom of the screen are the scroll bars, which allow you to move quickly and easily around your listing. These are used in exactly the same way as Workbench scroll bars. If you find them too fiddly, you can also move around using the cursor keys, so don't fret.

DIRECT MODE

If you press the escape key a completely new work screen will appear. This is called Direct Mode, and it acts on each command as you type it, rather than waiting for you to run the program. If you typed PRINT 12/17 in Edit mode, nothing would happen until you ran the program. It, however, you type it here, the command is executed immediately without affecting the listing in Edit mode.

Direct mode allows you to try out commands before they form part of your program, as well as carry out various house-keeping duties without disturbing the flow of your programming. If you wanted to see how many sprites or samples you had in memory, check how much disk space was available or see how two colours went together. This is the place to do it.

RUNNING A PROGRAM

To load a program, you need to click on the 'Load' option in the menu bar, and then choose the file using the selector (see panel). Once it has loaded you'll be presented with the complete listing. Now, to run it, all you need to do is press F1 or click on 'Run' in the menu bar.

To stop a program in its tracks, without waiting for the logical end, you need to hold down the Control key and press the C key at the same time. This aborts the current program and returns you to the edit screen. To see what I mean, load the 'Scrolling Text Demo' from your AMOS program disk, run it and then abort it.

MEMORY BANKS

AMOS is capable of some fairly nifty sprite and sample handling, but like any other program the data for these need to be in memory at all times, and saved with the basic program.

This is done by using the AMOS Memory banks, 16 blocks of RAM used specifically for resource data. Once something is loaded into a memory block, it is automatically saved with the program, and there's no need to reload any of it the next time you load. To see how it all works, go to Direct mode, and load the sprite file on this month's coverdisk by typing:

LOAD "KIKstrt.Abk"

Once the file has loaded, jump back to edit mode. Notice how there's no listing? So how do you check if AMOS has loaded anything? Simple. Go back to direct mode and type:

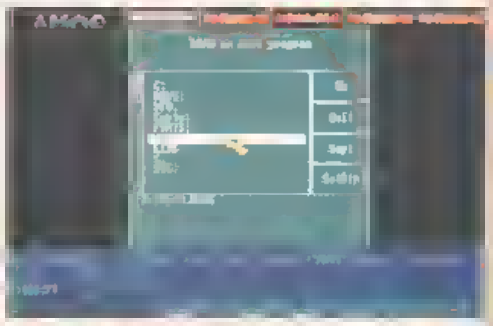
LISTBANK

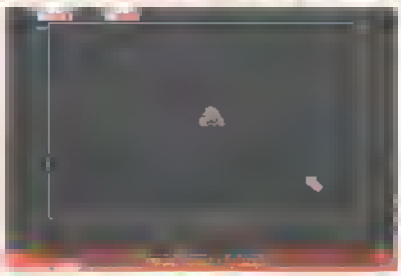
A list of the currently used memory banks appears, and there you should see Bank 1 contains five sprites. To see them in action, enter:

Sprite 1, 200, 200, 1

We'll deal with sprites in more detail later on in this booklet (page 18).

AMOS direct mode allows you to try out commands and do a little housekeeping without disturbing your programs.





VARIABLES

One of the most important things to know about when using AMOS is variables. A variable is a named area of memory used for storing information, such as a name, a number or any string of characters. In the case of the listing on this page, Answer$ is the name of the variable. Think of it as a pigeon hole called Answer$. The INPUT command tells the computer to put whatever you type into the pigeon hole marked Answer$, and whenever the computer looks back at that hole, it will be able to read your answer. A variable can be called almost anything you like, as long as you follow these rules.

- A name by itself means an integer variable (no decimal places). A, Answer and Total can all be used as integer variable names.
- A name with a # after it is a real number variable. The command A=10/4 would mean that A=2. A#=10/4 would give the result 2.5
- A name with a $ after it is a string variable, meaning that the variable will only have characters.
- A variable can't have the same name as an AMOS command. PRINT, NEXT and DIM can't be variable names, but PRT, NXT and D can.
- Variables are reset every time the program is started, so don't expect to see the same data the next time you load your file.
- A variable will only work in the part of the program it belongs in. To make a variable available for the entire program, including procedures, you need to make it Global. At the start of your program, include the command GLOBAL, and then the names of all the variables you want to use, separated by commas. For example.

GLOBAL Answer$, A, Hello#

THE MENU BARS

Do you really know what 'Block Hide' and 'Close All' do? If you're still a little confused by the mystic menu bars, then read on.

Those 10 words that you can see at the top of the screen constitute the menu bar, and that little box is going to make your programming life much easier, once you've got the hang of it. It contains 40 useful commands that let you do all sorts of system management tasks without touching a key. To use each one, all you need to do is move the mouse pointer so that it highlights the option you want to select, and then single-click on it with the left mouse button. Alternatively, you could just press one of the function keys. The top five options are selected using the keys F1 to F5, and the bottom row are selected using the keys F6 to F10.

THE DEFAULT MENU

This menu deals directly with the AMOS editor, and is on screen by default.

This is the menu that is on screen when the package loads, and when no keys are being pressed. It gives you access to two other menus, as well as giving you complete control over the editor. The commands are as follows:

F1: RUN: An obvious one really, this option runs the program currently displayed on screen. Before it runs it, it will test it for typing errors and similar bugs. If it finds any, it will alert you and abort the running.

F2: TEST: Like the Run option, this one checks the program for errors, alerting you as it finds them. As soon as it finds one, it stops the test and places the cursor next to the error.

F3: INDENT: To make your programs more readable, you might want to indent loops and procedures, making them easier to spot when scanning over the listing. Choosing this option automatically indents the program in memory.

F4: BLOCKS MENU: This option calls up the blocks menu, which we'll look at later.

F5: SEARCH MENU: Another menu that can be called from the default one. Again, read all about it later.

F6: RUN OTHER: AMOS allows you to hold two programs in memory at the same time. To run the other one, for example a sprite editor, use this option.

F7: EDIT OTHER: This option simply switches over between the currently displayed listing and any others that you might have stored in memory.

F8: OVERWRITE: This switches between the two editing modes. 'Insert' automatically makes room in the listing for anything you type, whereas 'Overwrite' writes over the current listing, replacing existing text with the new characters.

F9: FOLD/UNFOLD: This is used to hide procedures. If you have a particularly lengthy procedure which you find is slowing down your editing of the program, placing the cursor within it and pressing this key 'folds' it into memory, leaving only the title line of the procedure on display. To get your procedure back again, all you need to do is select this option again.

F10: LINE INSERT: This option creates a blank horizontal line at the current cursor position, making space for new lines.

THE SYSTEM MENU

The System menu gives you access to the floppy, as well as use of any accessory programs you may have loaded.

The System menu gives you access to the disk drive, and is displayed by holding down the shift key. With the shift key held, the function keys work as before.

F1: LOAD: Again, this one is self-explanatory: it loads a file from disk. You can then select the file using the file selector.

F2: SAVE: The opposite of load. Saves the current file to disk.

F3: SAVE AS: Lets you save the current file under a different name.

F4: MERGE: Inserts a program at the current cursor position without erasing the previous listing. With this, programs can be written in modules and then added together at the end.

MENU BARS

F5: MERGE ASCII: If you like you can write your AMOS listings using your favourite word processor, remembering to keep the line format the same. Save your document as an ASCII file, and then use this option to load it into the interpreter.

F6: ACC NEW/LOAD: Clears all the current accessories from memory, and loads all files off disk that have the '.ACC' extension.

F7: LOAD OTHER: This loads another program from disk and puts it in memory without displaying the listing. This is particularly useful for accessories such as the sprite designer, which it is always handy to have stored in memory.

F8: NEW OTHERS: Clears all accessories from memory. For accessories, read 'Programs not displayed in the edit window'.

F9: NEW: Clears the current program from memory. If the program isn't saved, the interpreter will ask you if you want to save it. Type 'Y' or 'N' to answer.

F10: QUIT: Exits AMOS and returns to the CLI. You will be prompted to save your program before the system exits.

ALTERNATIVE KEY SHORTCUTS

The AMOS edit window features a number of other keyboard shortcuts for menu selection. Here's the full list.

Amiga+L:	Load a program
Amiga+S:	Save a program
Shift+Amiga+S:	Save As
Control+B:	Block Start
Control+E:	Block End
Control+C:	Block Cut
Control+P:	Block Paste
Control+M:	Block Move
Control+S:	Block Store
Control+H:	Block Hide
Control+F:	Find
Control+N:	Find Next
Control+R:	Replace
Control+TAB:	Set Tab

THE BLOCKS MENU

The Blocks menu lets you manipulate large chunks of your program with ease, which is useful for tidying things up afterwards.

If you've ever used a word processor, you'll already be familiar with the principle behind 'Cut and Paste'. The Blocks menu lets you lift large sections from your listings and move them around using only a couple of mouse clicks. To show the blocks menu, hold down the Control key.

F1: BLOCK START: Marks the start of the block you want to highlight. Move the cursor in front of the first character and select this option.

F2: BLOCK CUT: Removes the highlighted block from the listing and stores it in memory.

F3 BLOCK MOVE: Moves the highlighted block to the new cursor position and deletes it from the old position.

F4: BLOCK HIDE: Deselects a selected block.

Once a block of text has been highlighted it can be lifted or removed at the touch of a button.

F5: SAVE ASCII: Saves the selected block as an ASCII file, which can then be loaded into any standard word processor.

F6: BLOCK END: Marks the end of the block. Move the cursor to the end of the block that you want to highlight and then select this.

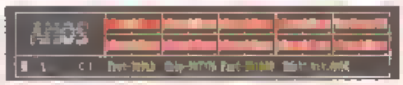
F7: BLOCK PASTE: Places a block stored in memory at the current cursor position.

F8: BLOCK STORE. Copies the block into memory but doesn't affect the listing.

F9: BLOCK SAVE: Saves the currently selected block to disk as an AMOS program file. The block can then be merged into another program.

THE SEARCH MENU

When you select the Find option from the search menu, this prompt asks you for the string to search for.



Where is that text you need? The search menu will look for it for you.

The search menu does exactly what you would expect it to do. It hunts through your listing for a pre-set string of characters, such as a variable name, and then does one of a number of things. To display the Search menu, hold down the Alt key.

F1: FIND: This option prompts you for a string of characters, and then searches down from the current cursor position until it finds a perfect match.

F2: FIND NEXT: The Find option stops when it finds the first match. This option prompts it to look for another match further along.

F3: FIND TOP: This is exactly the same as the Find option, only this one searches from the top of the listing regardless of the current cursor position.

F4: REPLACE. Searches through your listing for a match, and then replaces it with a second string input at the start of the search. If you have a variable name that you want to change, using this option is the easiest way of doing it. You will be asked to confirm each replacement.

F5: REPLACE ALL: Changes all copies of a word in your listing.

F6: LOW<>UP: Represents case sensitivity. In this mode, the search routine differentiates between upper and lower case characters. Clicking on this changes the mode to LOW=UP, in which upper and lower case letters are treated as identical.

F7: OPEN ALL: Opens all closed procedures in your program.

F8: CLOSE ALL: Closes all open procedures in your program.

F9: SET TEXT B: Lets you change the size of memory available for your listings. The more memory you have, the larger the programs you can fit into memory.

F10: SET TAB: This allows you to set the number of character spaces between each tabulation marker.

GAME DESIGN

Design is the backbone of any program. With a good design, the whole programming experience becomes much easier to handle. Here's what to look for.

Before any programmer can put his or her fingers to the keyboard, they must put pen to paper. A solid game design is the key to successful programming, as anyone who has tried to work without one will no doubt tell you. By figuring out right at the beginning how your program will work, what it will do and what it will look like, you'll save yourself all sorts of hassle later on in the project.

But, you might be asking, how exactly do I make a design? The first thing you need to do is work out exactly what you want to create, and then sketch a couple of notes. If, for example, you wanted to create a shoot 'em up, then you might write something like

"*Big Guns* will scroll both ways over a dozen levels, each one set on a different planet. The aim will be to shoot a certain number of alien eggs before they spawn alien ships. There will be four different weapon upgrades, ranging from a simple laser to homing missiles. At the end of each level, you'll be able to buy the weapons with your points – the higher your score, the better the weapons you can buy".

ON YOUR OWN TERMS

OK, so that's your brief. Now you need to think about that in programming terms. How will you make the game scroll both ways? How can you tell when the eggs are ready to gestate? How are the different weapons going to be represented? All these decisions must be made early on, just for the sake of practicality. Some things might not be possible without a loss of speed or playability, and correct planning will ensure that you spot these things early on.

RUNNING ORDER

Now comes the tricky part. You have to work out a running order for the program listing itself and begin to formulate the routines and patterns involved in making your program work the way you want it to. How are you going to get the ship and the backdrop scrolling? How many times a second should you read the joystick? That sort of thing. It might seem daunting, so let's break it down into smaller programs.

Figure out the various components of your game, and work on each seperately. In the case of the scrolling, your notes would look something like this

```
(Variable D=Scrolling direction (1=left, 2=right)
display screen
check D
adjust screen position accordingly
display screen again
```

This may not sound much like a guideline for a game, but it's only preparation for a flowchart.

FLOWING NICELY

As a rule, flowcharts are impossibly dull to create. However, they are also invaluable when writing a game. If you are creating an especially long listing, it's handy to have a list to refer to in order to stop you from getting lost.

A flowchart breaks a program down into single steps, making the entire challenge much easier to cope with. In a game like the one that we have referred to above, writing an engine that moved the backdrop and kept track of all the sprites on screen at once from scratch would be too much for the beginner. Working from a well-written flowchart makes it simple enough for even a Megadrive owner to get to grips with.

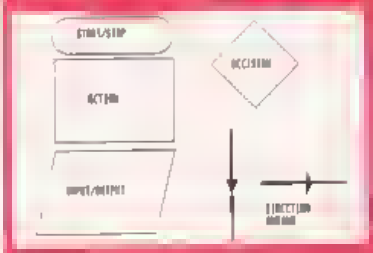
To design your flowchart, you need to break your program down again, into the smallest lumps you can. Remember, the more you break it down now, the less you have to figure out later on.

OTHER POINTS TO CONSIDER

The listing itself isn't the only thing that needs a lot of thought put into it. You should also spend some time working out how the game will look and sound. Can you draw well? Are you able to compose a suitable soundtrack for the game?

It's best to be honest with yourself – if you don't think you're up to a certain task, then find someone who is. It makes all the difference in the end.

FLOWCHART SYMBOLS



Although you don't have to follow these old-school flowchart styles, it can make things a lot easier, for you in the long run. Here are the five main symbols you will find yourself working with.

1) ACTION

This represents anything done within the computer, such as calculating variables, reading data or setting up a screen.

2) INPUT/OUTPUT

This represents any form of input or output, from displaying something to reading the joystick.

3) START/STOP

This one is obvious – I don't really need to tell you what it is for.

4) DECISION

For those moments when the program can go one of two or more ways, a decision box shows the possible routes. Write the questions in the box, and label the exit arrows where possible.

5) ARROWS

The simplest arrows show the flow of the program, and it's vitally important that you mark the direction on them; without them, routines such as procedures become impossible to understand.

GAME DESIGN

WE ALSO STOCK ASSASIN GAME (1-40)LSD CHEATS (1-40) FRED-FISH, TBAG AND MUCH MORE

"a flexible and well presented service". ..Mr J K. from Cotteridge
"at what Amiga DTP users have been waiting for". .Mr D.T. from Leeds
"thankyou for the efficient service" Mr S T. from Sheffield.
"exactly the service I was looking for". Mr A.D from Thetford
"thankyou for your excellent service". .Mr D T. from London E11

SELECTAFONT COMPANY

Over 1200 Scalcable Fonts available for the Amiga in both Compugraphic Style and Adobe Types. Select from over 60 Disks. Each Disk contains 12 Fonts and costs just

£3:00 each.

BONUS DISKS

Please send a large S.A.E with 28p Postage for a Print-Out of all Fonts. Also state which Programs you will be using them in

(DEPT CU), 24 Thorpe Road Hawkwell Nr Hockley Essex SS5 4JT

need we say more?

Probably the finest Collection of Fonts in the U.K.

SELECTAFONT NEWS CORNER:

1: 250 NEW CG FONTS direct from the USA
2: ALL FONTS now available in BOTH FORMATS
3: FINAL COPY 2 (update) COMPATIBLE

The Official AMOS Public Domain Library

1 Penmydd Road Penlan Swansea SA5 7EH
☎ /Fax 0792 588156 Run by Anne Tucker

AMOS PUBLIC DOMAIN
NOW OVER 470 DISKS AVAILABLE
UK....£2.00
ROW & EUROPE £2.25
Multi disk programs are charged per disks not per title
BUY 10 DISKS = GET ONE FREE!
10% Discount for TA Subscribers
SAE FOR MORE DETAILS

UK & EEC prices include VAT @ 17.5%
ROW prices exclude VAT
All overseas orders include extra postage

Cheques/Postal Orders should be crossed and made payable to AMOS PD LIBRARY
Overseas orders should be paid for in STERLING ONLY
by cheque drawn on a UK bank or by International Money Order
THANKYOU

TOTALLY AMOS

This is the magazine disk for all Amos users which has been approvedby Europress Software. Here you can meet with other AMOS enthusiasts to learn with other readers
Each disk comes packed FULL of programs and routines which you can load into AMOS and see just how they work, also we have articles and tutorials on lots of AMOS related topics
Hopefully there is something for everyone in each issue of Totally Amos, so join the TA's and see the world of AMOS Programming

PRICES: UK - £2.50 per issue
OR £15.00 for 1 year (6 issues)
Please state at which issue you would like your subs to start from.
ROW £3.00 per issue OR £18.00 (air mail) for 6 issues
BACK ISSUES ALWAYS AVAILABLE
Sample issue available as APD 341 Price £2.00

ISSUE 10 AVAILABLE IN MAY

Amos in Action

By Anne & Len Tucker
Learn how to write a maze game like Marvin the martian step by step The book has a coupon to send for a free disk £12.95

CATALOGUE DISK, updated monthly
£1.00 UK £1.50p Row

FOR MORE INFORMATION PLEASE SEND A STAMPED ADDRESS ENVELOPE TO: ANNE TUCKER AMOS PD, 1 PENMYNYDD RD, PENLAN, SWANSEA, SA5 7EH

NAME..... ADDRESS.....

YOUR FIRST STEP

Now that you understand how to transfer your design into a program flowchart, you're ready to program. Here we go...

Before you can really piece together an AMOS program, you need to spend a little time familiarising yourself with the fundamental components of the language and its construction. I know it sounds like you need to do a degree course, but believe me it isn't that bad. These pages outline the basics of AMOS construction, which you'll need if you're to turn your flowchart into a fully functioning program.

VARIABLES

Any piece of information which is stored and used again (a player's name, the number of ships left in a shoot 'em up etc) is stored in memory and labelled. This is known as a variable, so called because the information can vary but the name remains the same. To assign information to a variable, we use the command Let, like this.

```
Let MAG$ = "CU Amiga"
```

Mag$ is the name of our variable in this case and we are filling it with the name 'CU Amiga'. Now we have that information stored, anytime we want to use that name, we can call Mag$. Here's an example

```
Let MAG$="CU Amiga"
Print "My favourite magazine is ";Mag$
```

See how it works? You can change the information in a variable to almost anything you like – try it. Change the information, and run the program again. Now add these lines to the start of the program, before all the others

```
Let MAG$="Homes And Gardens"
Let MAG$="MCN"
Let MAG$="Creative Cricket"
```

What will happen if you run the program now? Rnn it and see. The contents of a variable can be altered as many times as you like, but the program will always replace the old information with the new.

When naming a variable, a couple of rules need to be followed. Firstly, no two variables can have the same name. Secondly, a variable can't contain the same letters as a program instruction (Print, Run, Draw). Thirdly some variables need an extender on the end of the name. Mag$, for example, is a String variable, denoted by the dollar sign ($) at the end of the name. You can put anything you like in a string variable, but bear in mind that any numbers you store here are stored as characters rather than mathematical symbols – you won't be able to use them in mathematical terms. All strings are enclosed in quote marks (""), without them you'll get an error message.

There are two other kinds of variable recognised by the system. The first are Integers (Whole numbers). These have no extender after the name, and only numbers can be stored in these. If you try to store a string of characters, you'll get an error message. The other kind are Real numbers, which allow decimal places, unlike Integers. A real number is recognisable as having a hash (#) after the name. With that in mind, can you tell which of these are legal and which aren't? Try them and find out.

```
Ed$="Dan Slingsby"
Age="21"
Time=12.50
Time$=12
Nna=60
Pressies#=3.14159
```

CONTROL ROUTINES

AMOS contains a variety of different commands for controlling the flow of your program, which range from simple directions to condition testing and directing a program depending on the outcome of a variable. These will probably seem a little complicated at first, but try them out a few times, and you'll find them a lot simpler to use than an equivalent program that doesn't use them!

FOR...NEXT

If you have a segment of program that needs to be repeated a certain number of times a simple loop is the easiest way to do it, rather than write out the same piece of code over and over again. If you wanted to print your name 20 times, you could write:

```
Print "My Name"
Print "My Name"
Print "My Name"
```

and so on, but surely it would be far easier to use something like

```
For A=1 to 20
  Print "My Name"
Next a
```

'A' is a variable, and can be anything you like. See how it works?

DO...LOOP

If you have a piece of program that you want repeated indefinitely, looping forever, then a Do... Loop loop is all you need. Do marks the start of the loop, and the Loop command tells the program to go back to the Do instruction.

```
a=0
Do
 Print A
 a=a+1
Loop
```

REPEAT...UNTIL

Let's assume with the program above that you want it to count to a present random number. There are two ways to do this – one is to do a For Next loop with a random number in the For instruction. The other is to do a Repeat., Until, where the program will break the loop once a condition has been met. Try this

```
Z=Rnd(30000)
a=1
Repeat
```

Three diagrams to show how the different control structures work. The Go-Loop routine will run forever, but the For-Next and the Repeat-Until loops will run until a specified condition has been met such as a score reaching a certain level.

```
Print A
A=A+1
Until A=Z
Print "Found it At Last"
```

IF...THEN...ELSE

Condition testing is the heart of programming. An If...Then instruction is the heart of decision making – we do it every day. IF it's warm THEN don't wear a coat. That sort of thing. In the programming sense, it works in exactly the same way. Try this:

```
a=0: B=Rnd(9)+1
Repeat
 Print "Give me a number between 1 and 10"
 Input A
 If A=B then print "Correct!":Z=1
 If A<>B then print "No, sorry"
Until Z=1
Direct
```

The three condition testing symbols are:

'='	Equal to
'<'	Less Than
'>'	Greater than
'<>'	Not equal to

Combinations of these can be used (provided that they don't contradict each other – something can't be equal and not equal!) in any of the condition tests.

If you like, you can extend the instruction to include 'Else'. This tells the machine what to do if the condition isn't true. With this, our new program would look something like:

```
z=0: B=Rnd(9)+1
Repeat
 Print "Give me a number between 1 and 10"
 Input A
 If A=B Then z=1 Else Print "Sorry, try again."
Until Z=1
Print "Well done!"
Direct
```

WHILE...WEND

A While...Wend loop is similar in principle to a Repeat...Until loop in that it waits for a condition to be met before it breaks the loop. The instruction While is followed by the condition, and Wend signals the end of the loop. For example:

```
X=0
While x<20
locate x,0
print "*"
X=X+1
wend
Direct
```

TABLE 1

```
Do
  Cls
  Print "1) Option 1"
  Print "2) Option 2"
  Print "3) Option 3"
  Print "4) Goto Editor"
  Print "5) Goto Direct"
  Input A
  On A OP1,OP2,OP3,Edit,Direct
Loop
Procedure OP1
  Cls
  Print "You chose Option 1"
  Wait Key
End proc
Procedure OP2
  Cls
  Print "You chose Option 2"
  Wait Key
End proc
Procedure OP3
  Cls
  Print "You chose Option 3"
  Wait Key
End proc
```

END/EDIT/DIRECT

These are used to end the program. The first, End, just stops things in their tracks, and asks you which mode to go to. Edit ends the program and goes straight back to Edit mode, and Direct ends the program and goes straight to Direct mode. To see how they work, replace the 'Direct' command at the end of the last program with 'End' or 'Edit'.

ON...PROC/GOTO

On... is a very powerful command indeed. It works with an integer variable to determine where the program should branch to. An example of this is a menu screen. If you wanted, you could just put:

```
If a=1 then PROC1
If a=2 then PROC2
```

and so on. Or you could use a command like:

```
On A PROC1, PROC2, PROC3, PROC4...
```

Try the listing in Table 1 to see how it works.

ARRAYS

Variables can be clustered together in Arrays like this one

Let's say you are dealing with a large set of variables in the same format – as in a database. You could name each variable separately, (X1, X2, X3, X4, X5), but that wastes a lot of memory and makes the program hard to follow. What you need is an array – a collection of shoe boxes stuck together. An array has a single name, and the contents are called using a co-ordinates system. Arrays can be in as many dimensions as you want, from two to 19 dimensions.

DIM

Dim creates a new array ready for filling. DIM CU(19) creates an array with 19 spaces. DIM CU (10,10) creates an array with 19 main spaces broken into 10 more, giving 100 spaces.

READ, DATA

Filling an array by hand can be time-consuming. If you use Let array(1)=XXX, Let array (2)=XXX etc. The simplest way to do it is to set up a Read statement and a collection of data. Read reads the next item of data and puts it in a preset position, as shown here:

```
Dim STAFFS(7,2)
For name=1 to 7
  For place=1 to 2
    Read Staff$(name, place)
  Next place
Next Name
Data "Dan","Editor","John","Dep
Ed","Nick","Tech
Ed","Gordon","Design","Mat","Tech
Advice","Tony H","Star Fighter","Tony
D","Freelance"
```

Run the program and nothing will happen. What you need to do now is check that the array has been filled. Add the following lines to the program:

```
For Name=1 to 7
   Print staff$(name,1),staff$(name(2)
Next Name
```

Handy, isn't it?

WORKING WITH TEXT

To begin with, most of your programs will probably involve a lot of text manipulation, and there are few packages than can handle this better than AMOS.

Practically every program you ever write will include some text, whether it's just your name scrawled on the title page or a complex parser for an adventure game. Working with text is one of the easiest things that you can do with AMOS, which is why most people's first program involves writing their name in random colours all over the screen. Here are the main text commands used by AMOS and some examples of how to get the best out of them.

PRINT

The first command you need is Print, which obviously prints something to the screen. It always prints at the current cursor location, and works in two ways. If the command is followed by a string of characters enclosed in quote marks (""), it will print the contents of the quote marks only. For example:

Print "Hi Dan" will print Hi Dan
Print "12*7" will print 12*7.

If you take away the speech marks, however, something totally different happens. Instead of printing the entered characters, the program will look for a variable in that name, or if you have entered a mathematical operation, it will print the answer. In our examples, the program would look for a variable called 'Hi Dan', and would also print 84.

LOCATE

So far, whenever you have used a print statement, it has always printed in the top left-hand corner of the screen or down the left side. So what happens when you want to print in the middle of the screen? I'll give you a clue. The text always prints at the current cursor position. Give up? You move the cursor. There are two main ways to do this, the easiest being to use the Locate command.

To use the Locate command, all you need to do is specify where you want the cursor to move to, using two co-ordinates, the first to specify the X (across) position, and the second to specify the Y (down) position. For example:

```
LOCATE 17, 10: PRINT "HI DAN".
```

will print a message to the Ed 'slap'bang in the middle of the screen.

CMOVE

CMOVE is short for Cursor Movement and is the other main way of shifting the cursor position. Instead of nominating an absolute position via co-ordinates, CMOVE works by moving the cursor relative to its current position. Again it uses a set of co-ordinates, which are added to the current cursor co-ordinates. Positive numbers move the cursor to the right and down, and negative numbers move the cursor left and up. Try this example:

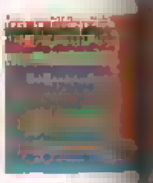
```
Locate 17,10
Print "HI DAN"
Cmove -6,-2: Print "Above"
Cmove -6, 4: Print "Below"
```

See how it works? Experiment with different co-ordinates in the Locate command to see the benefits of the Cmove command.

PEN

The Pen command changes the colour index of the text printed on screen. Depending on your screen mode the index numbers can run from 0 to 63. All subsequent text will be printed in the selected colour until another Pen command is used. Try this example:

```
For P=0 to 15
  Pen P
  Print "This is Pen" ;P
Next P
```



PAPER

The Paper instruction works in the same way as the Pen command, only this time it changes the background colour beneath the text. If you imagine that each character is a letter on a typewriter, then you'll know that there is a square of metal around each letter. It's this area that the Paper instruction changes. Try this program to see what I mean:

```
For P=0 to 15
  Paper P
  Print "This is Paper" ;P
Next P
```

INVERSE, SHADE, UNDER

Inverse mode is when the Pen and Paper colours are reversed, creating a negative of the text. This type would be white ink on black paper when inverted. Shade mode darkens the text slightly to highlight it. Underline mode draws a line underneath your text. All three modes are switched on using XXX On, and switched off with XXX Off (replace XXX with the appropriate command). Try this program to see how it works.

```
For P=0 to 13
  Paper P: Ink P+5
  Print "Text in normal mode."
  Inverse On
  Print "Text in inverse mode"
  Inverse Off
Next P
```

Replace the Inverse command with the Shade command, and then try it together with the Under command.

MOVING ON

You now know enough to start writing your own programs. Try to write a program that asks you for your name, and then prints it all over the screen in a variety of styles and colours. Once you can do that, you're ready to tackle the next stage.

TEXT

17 BIT SOFTWARE Est. 1988

PUBLIC DOMAIN SOFTWARE AT ITS BEST!

WE STOCK THE LOT!
FISH TO 810!
AMOS DISKS!
TBAG DISKS!
NZ DISKS!
AMICUS!
AMIGAN!
ALL CLR TITLES!

FAST SAME DAY SERVICE, HELPFUL SALES STAFF, ESTABLISHED FOR OVER 5 YEARS
40,000 MEMBERS THROUGHOUT THE WORLD, WELL OVER 4000 TITLES IN STOCK!

HOW TO ORDER

BY PHONE
(0924) 366982
CREDIT CARD / SWITCH

BY FAX
(0924) 200943
WITH ORDER & CREDIT CARD DETAILS

BY POST
PLEASE MAKE CHEQUES PAYABLE TO:
17 BIT SOFTWARE
1ST FLOOR OFFICES
2/6 MARKET STREET
WAKEFIELD
WEST YORKSHIRE
WF1 1DH

OFFICE HOURS

MON-THURS 9.00 TO 6.00
FRI & SAT 9.00 TO 5.30
WE ARE OPEN TO PERSONAL CALLERS FROM 9.00 TO 5.30.

DISK PRICES

17 BIT, FISH ETC	£1.25
SCHEME 17	£2.00
AM/FM MAG	£2.50
AM/FM SAMPLES	£2.50
CLR SINGLE TITLE	£3.50
CLR 2 DISK SET	£4.50
CLR 3 DISK SET	£4.99
CAT DISKS	50P

POSTAGE RATES

UK PD ORDERS	50P
OVERSEAS ORDERS	20%
(MIN OVERSEAS P&P	1.00)

PLEASE ADD 75P P&P FOR COMMERCIAL GAMES, DISK BOXES ETC

BUY 10 DISKS AND GET 1 EXTRA DISK FREE!
BUY 20 DISKS AND GET 3 EXTRA DISKS FREE!

PLEASE NOTE: (AB) AFTER A DISK MEANS 2 DISKS ETC. PLEASE STATE THIS WHEN ORDERING. THANKS

NEW FOR CDTV!

The 17 Bit Collection
Over 1600 disks worth of the best in public domain games & demos Etc. All on a double CD! Hundreds of Demos, Pics, Games & Utils for only £40.99 including P&P! Unbeatable Value!

CDPD VOLUME #1
Containing Fred Fish disks from F001 to F660!
Well Worth £19.99 + 75p P&P!
Thats over 650 disks Worth!!

CDPD VOLUME #2
Continuation of Fred Fish from F661 To F750 + The Entire SCOPE & JAM Ranges! Another packed CD for only £19.99 + 75p P&P

DEMO CD VOLUME #1
For the Connoisseurs, this CD contains Demos, Intros, Cliparts, Modules, Samples etc!
Only £19.99 + 75p P&P!

Commercial Stash!

Get Some Real Gaming Action!

Amos 3D	£18.99
Amos Compiler	£18.99
Captive	£12.99
Defender Of The Crown	£5.99
Dream Team Compilation	£17.99
Harlequin	£14.99
Lemminge 2	£22.99
Lionheart	£21.99
Max Compilation	£12.99
Megalomania	£11.99
Myth	£14.99
No Second Prize	£17.99
Nigel Mansell World Cup	£17.99
Putty	£18.99
Realms	£11.99
Streetfighter II	£18.99
Spritz (Art Package)	£9.99
ThunderHawk	£17.99
Wing Commender	£18.99
Zool	£17.99

HOT TEAM 17 SOFTWARE!
Choose a Free PD Disk! With Any Team 17 Title

Alien Breed Remix	£9.99
Assassin	£18.99
Body Blows	£18.99
Project X	£17.99
Superfrog	£18.99

Bits 'N' Bats

"SPACE WARS"
See the latest in Amiga Animation on VHS Video! 24 bit dynamic Hi-Res movie from T. Richter
Only £11.99 + 75p P&P

"HOBBITS & SPACESHIPS"
Ex Crusader Bjorn Lynne brings you the latest in Synth Sounds on Audio CD. 10 superb tracks running for 72 mins for only £12.99 + 75p P&P

"THE FINAL FRONTIER"
A 4 disk mag which no Trekkie should be without. Includes exclusive artwork by T. Richter and upto date seminar news etc. Only £6.95

"AM/FM"
Issue 11 of this ever popular disk mag for music enthusiasts is now available Only £2.50 or £5.00 with sample disk. Back issues also available!

LSD "LEGAL TOOLS!"
We currently stock all the LSD legal tools utility compilations from 1 to 80! A catalogue is available of these titles for £1.00 inc P&P or free with orders of 10 disks or more

"LSD GRAPEVINE #14"
As always, our most popular disk mag is packed with controversial topics and news from the 'Scene' Dont miss it at £3.75!

Attention Overseas Traders!
We are currently looking for reliable companies to represent us in the overseas market. If you would like to be an official 17 Bit Dealer, and would like to know how YOU can benefit Contact us by Tel or Fax NOW!
Join The Leaders In Public Domain!

ASSASSINS GAMES DISKS!
If you thought the first 30 were good, you should see the next lot! Now a total of 56 issues available at incredible prices! Have a look below!
Any 10 for £11.99 Any 20 for £21.99
Any 30 For £29.99 Any 40 for £37.99
Any 50 for £45.99 or just take the whole lot for an incredible £49.99!

LATEST DISKS!

+2503	Windblend A1200 FracGen
+2502	More 1200 Only Utils
+2501	Childrens Songs
+2500	Gladiators Music Disk
+2499	Fruit Salad Game
+2496	Picture Puzzle
+2497	Marcos Box Of Fun (1.5MB)
+2496	Excell Database
+2495	Attraction Music Disk
+2494 (AB)	18th Hole Golf Game
+2493	The Enforcer Anim
+2492	Ham-8 Vistas
+2491	Vidi Effects
+2490	Drum Loops/Samples
+2489	Cindy Crawford Slides #2
+2488	Cindy Crawford Slides #1
+2487	Tad & Medgads Menus
+2486	Menu Launcher
+2485	PolyED V1.0
+2484	The Money Program
+2483	Cli-Tris & Pacatac
+2482	Hillbilly Anim
+2481	Bagman
+2480	Amy PD Review V1.1
+2479	Singalong Nursery Rhymes
+2478	How The Earth Began
+2477	Flashback Demo
+2476	Bomb Jacky
+2475	Create Adv. Games V2.1

NEW CLR TITLES!

CLU11	CALC V1.3
CLU12	Virtual Windows V1.0
CLU13	Datos
CLU14	Stock Controller
CLU15	Epoch V1
CLE10 (3)	Basically Amiga
CLE14 (3)	Ecology
CLE15	Fastfrol II
CLE16 (2)	Kings & Queens
CLE17	Thingamajig
CLE18	Work & Play
CLE19	Play It Safe
CLG18	Skin 29
CLG20	Jungle Bungle
CLG21	Flower Power
CLG22	Stocking Fillers
CLG23	Marvin The Martian
CLG24	Easy Money

SCHEME 17!

SS23	Personality Analysis
SS22	Lockout (H.D Security)
SS21	Mindmatch
SS19	Techno Attack IV
SS18	Pools Predictor
SS17 (3)	Finlandia - £5.95
SS16	Freakscene #1
SS15	Kids Karaoke
SS14	Crystal Symphonies II
SS13	Techno Attack III
SS12	Christmas Karaoke

INTERACTION

A game often stands or falls on its control method. On these pages we show you how to ensure that yours is a winner.

You already know one method of entering data from the keyboard – the Input command – but there are a variety of others which can be used to far greater effect.

INKEY$

Inkey$ tests the keyboard to see if a key is being pressed, and enters it directly into a variable. Whereas Input requires a 'return', Inkey$ works immediately. If no key is pressed, the instruction leaves the variable blank and carries on. See Table 1.

TABLE 1

```
Do
  X$=Inkey$
  Print x$;
Loop
```

But what if you want the program to wait until you have pressed a key? One way to do this is to stick a small bit. .Then line in the program. See Table 2

SCANCODE

Scancode is used to check the internal number for any of the keys on the Amiga keyboard. Including the ones with no visible effect, such as 'Help' or the function keys. See the example in Table 3.

Once you have your Scancodes, you can use them in conjunction with the Key State command, which tests whether a key is currently being pressed. If the Key State test is true, it will return a result of True. To see what I mean, have a look at Table 4.

INPUT$

Input$ is a different command to Input, so read this carefully. Input$ asks for a set number of characters, and places them in a nominated variable. See Table 5 for an example.

TABLE 2

```
Do
  z=0
  Repeat
  x$=Inkey$
  If x$<>"" then z=1
  Until Z=1
  Print x$;
loop
```

TABLE 3

```
Do
  While K$=""
  K$=Inkey$
  Wend
  If Asc(k$)=0 Then Print "You
Pressed A Key With No ASCII Code!"
  Print "The Scancode is ";Scancode
  K$=""
Loop
```

TABLE 4

```
Do
  If Key State (69)=True Then Print
"You've Escaped!"
  If Key State (95)=True Then Print
"No I won't help you!"
Loop
```

WAIT KEY

This instruction pauses the program until any key has been pressed.

STICK AROUND

Reading the joystick in AMOS is something that you will inevitably be using a great deal so you'll be happy to know that there are a few simple commands that make this just as easy as printing your name. The commands are as follows:

JLEFT, JRIGHT, JUP, JDOWN, FIRE

These five commands check if the various directions (counting the fire button as a direction) are being used, returning a value of 1 if the test is true. The number in brackets which follows the instruction is the number of the port under test. See Table 6.

TABLE 5

```
Clear Key: Rem Clears keyboard
buffer
Print "Please type 10 letters"
c$=Input$(10): Print "You typed ";c$
```

TABLE 6

```
Do
 If Jup(1) Then Print "Up"
 If Jdown(1) Then Print "Down"
 If Fire(1) Then Print "Fire"
Loop
```

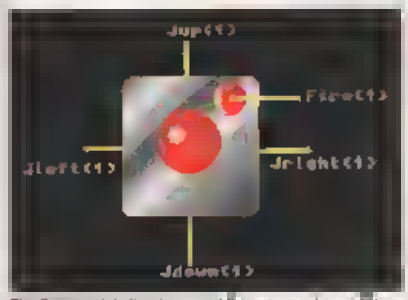
MOUSEY MOUSEY

The Amiga is perfectly suited to mouse-controlled games, and AMOS is more than capable of creating those games.

HIDE/SHOW

These two commands are used to hide and redisplay the mouse pointer for joystick controlled games such as shoot 'em ups. Use the command Hide On to remove the pointer, and Show On to redisplay it.

INTERACTION



The five joystick directions and the commands used to read them. Remember, the number in brackets is the joystick port.

CHANGE MOUSE

This instruction lets you change the graphic used for the mouse pointer. There are three pointers in memory at all times, so the command Change Mouse 1, 2 or 3 changes the pointer to an arrow, a crosshair or a clock. Any number higher takes a graphic from the sprite bank. Your only limit is that the graphic can't be any more than 16 pixels wide or have more than four colours.

MOUSE KEY

This checks the status of the mouse buttons, and returns a bit-pattern. To see the bit pattern for the items available, try the program in Table 7.

TABLE 7

```
Do
  Locate 0,0
  M=Mouse Key: Print "Bit Pattern
Number ";M
Loop
```

To do a one shot test of the mouse buttons, to see if a button has been 'clicked' use the Mouse Click command instead.

X MOUSE, Y MOUSE

These two commands fill double functions, depending on the way they are used. In the format 'Variable=X Mouse', the current X hardware co-ordinate (which isn't always the same as the screen co-ordinate) of the mouse is stored in a named variable. This is useful for testing where the mouse is.

By inverting the command and using it in a different way ('X Mouse = 100'), you can set the X hardware co-ordinate, thereby moving the mouse to a new position. See in Table 8.

LIMIT MOUSE

Normally the mouse has the run of the screen, but you can limit its movements to a rectangular portion by defining the top left and bottom right corners of the box of hardware co-ordinates. Try this program.

```
Limit Mouse 50,50 To 300,200
Wait Key
```

TABLE 8

```
Do
  X=X Mouse: Y=Y Mouse
  Locate 0,0: Print "X:";X;" Y ";Y
  If Mouse Click then X Mouse=Rnd(320) Y Mouse=Rnd(200)
Loop
```

MENUS

Menu bars are something we all take for granted – anyone who has had more than a week with an Amiga knows that holding down the right mouse button makes a line of menu options appear. With that in mind, one of AMOS's strongest points is its ability to build large and complex menus with minimum fuss.

MENU ON

Turns on the menu bar. Don't bother doing it at the moment, because you haven't defined a menu yet. To do so, you need to use the Menu$() instruction. This works in two ways. The first is to have a single figure within the brackets, which defines a title for the menu bar. Therefore:

```
Menu$(1)="About"
Menu$(2)="Options"
Menu On
```

Creates an active menu bar, but with no options. You need to create the options with the second use of the Menu$() instruction. This time you use two or more figures between the brackets, separated by commas. The first figure shows which menu heading the menu option appears under, the second is the order the item appears in, the third (if there is one) puts the option on a side branch menu. Add the lines shown in Table 9 to the program. Now run the program and see how it works.

CHOICE

The Choice() instruction is used to see which menu option you have chosen. The instruction 'head=Choice' will read the menu heading number into the variable 'Head'. To read the menu option chosen, you need to number the Choice command. Add the lines in Table 10 to your listing to see what I mean.

TABLE 9

```
Menu$(1,1)="About Menus"
Menu$(1,2)="About CU"
Menu$(2,1)="New Game"
Menu$(2,2)="Old Game"
Menu$(2,3)="Quit"
```

ON MENU PROC

Instead of writing out a whole string of commands every time you want to read the menu, you can assign a procedure to each of the menu titles using the On Menu Proc instruction in conjunction with the On Menu On command. This system checks the menu bar 50 times a second without any programmed checks by you, so your program can continue as normal. Try the listing in Table 11.

Note: Once 'On Menu Proc' has been used, the On Menu On system stops, so remember to put an 'On Menu On' at the end of each procedure.

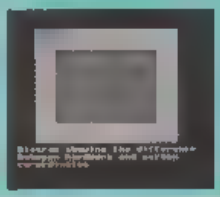


Diagram showing the difference between hardware and screen co-ordinates

Hardware co-ordinates refer to the entire screen, not just what's visible as this diagram shows.

TABLE 11

```
Menu$(1)="Mouse":
Menu$(2)="Quit"
Menu$(1,1)="Arrow":Menu$(1,2)="
Cross":Menu$(1,3)="Clock"
Menu$(2,1)="Editor":Menu$(2,2)="
Direct"
Menu On
On Menu Proc MSE, QWIT
Rem: Do something
Do
 For x=1 to 100
 print X;
 Next X
Loop
Procedure MSE
 If Choice(2) then Change Mouse
Choice(2)
 On Menu On
End proc
Procedure QWIT
 If Choice(2)=1 then Edit
 If Choice(2)=2 Then Direct
 On Menu On
End Proc
```

TABLE 10

```
Do
  If Choice and Choice(1)=1 and Choice(2)=1 Then Print "This is a menu option"
  If Choice and choice(1) =1 and choice(2)=2 Then Print "What do you want to know?"
Loop
```

BASIC GRAPHICS

Bars and bosses like these can be used for all sorts of applications, from the prompt windows or selection boxes.

Before you dive headlong into the slightly difficult world of sprite and blitter objects, why not play around with AMOS's more fundamental graphic tools?

AMOS has a large collection of tools for defining open and closed polygons and other geometric shapes. All are based on a simple co-ordinate system, with the first figure marking the position across from left (0) to right (320) and the second figure marking the position from the top (0) to the bottom (200 NTSC, 256 PAL).

RAINBOW BRITE!

Before you draw anything you need to choose your colours. AMOS has a few simple, but effective instructions for palette selection. One thing to note here is the colour index syntax. This is the name given to the settings of the individual colours.

A colour index is a three-figure hexadecimal figure which tells the processor how much red, green and blue should be mixed to create the colour – just like using the colour mixer on a program such as *Deluxe Paint*. Take a look at the quick decimal/hex conversion table below.

INK

The Ink command is used to set the colour for subsequent drawing operations, and works in exactly the same way as the Pen command. See Table 1.

TABLE 1

```
I=4
For A = 10 to 100 Step 10
 For B = 1 to 10
  Ink I
  Draw 10, A+B to 100, A+B
  Next B
 I=I+1
Next A
```

COLOUR

This instruction sets a colour in the palette to a specified colour index. Add these lines to the start of the previous listing, and see if you can guess what colours will appear.

```
Colour 4, $0F0
Colour 0, $111
Colour 2, $00D
Colour 7, $123
Colour 6, $AF0
Colour 8 $002
```

PALETTE

Obviously, keying in all those Colour commands every time you want to change the palette is going to get very tiresome, with the Palette command you can change all the colours at once by typing a string of colour indexes separated by a comma. If you don't want to change a particular colour, then just leave a space between that pair of commas.

PLOT

Plot colours a single pixel on screen using the current Ink colour. For example:

```
Ink 1 0
Plot 100,100
```

DRAW/POLYLINE/POLYGON

The Draw command draws a straight line between two points. Both points can be set, or you can leave it to draw from the current cursor position. To see what I mean, go to direct mode, clear the screen using CLS and type the following line.

```
Draw 100,100 to 200,100
```

Bingo, a line appears. Now try the next line:

```
Draw to 80,80
```

See how the line has automatically been drawn from the end of the last one? Incidentally, with irregular multiple line shapes, such as polygons, the Polyline command works like an extended Draw instruction, in that you can stick as many 'to X,Y's as you like on the end. To draw a filled polygon, use the Polygon command. For example:

```
Polyline 20,20 to 100, 100, to 80, 150 to 20,140 to 20,20
```

draws an empty polygon.

```
Polygon 20,20 to 100, 100, to 80, 150 to 20,140 to 20,20
```

draws a filled one. Easy!

BOX/BAR

If you want to draw a hollow box on screen the easiest way to do it is to use the Box instruction. Like the Draw command, two sets of co-ordinates are used – these ones specify opposing corners of the box.

DECIMAL/HEX CONVERSION TABLE

Decimal:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hex:	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

To enter hex numbers in AMOS, you need to add the prefix $. For example, to enter the number 15 in hex, you would type $F. In the colour index, the three digits correspond directly with the red, green and blue settings. Therefore absolute red is $F00, a medium grey is $777 and white is $FFF. Got that?

BASIC GRAPHICS

GASTEINER

Unit 2
Millmead Business Centre
Millmead Road
London N17 9QU
Tel: 081 365 1151
Fax 081 885 1953

ALFA DATA ICD Marpet ~~BMC~~ POWER ATARI vortex PHILIPS

AMIGA COMPUTERS

[illegible]+	£269.00
[illegible]	£269.00
[illegible] 20Mb Hard Drive	£465.00
[illegible] 60Mb Hard Drive	£549.00
[illegible]200	£379.00
[illegible]200 20Mb Hard Drive	£529.00
[illegible]200 40Mb Hard Drive	£599.00
[illegible]200 65Mb Hard Drive	£639.00
[illegible]200 80Mb Hard Drive	£679.00
[illegible]200 120Mb Hard Drive	£769.00
[illegible]00 120 Mb Hard Drive	£2089.00

MONITORS

[illegible]ips CM8833 Mk2	£229.00
[illegible]mmodore 1084/5SD1	£209.00
[illegible]mmodore 1960 Multisync	£439.00
[illegible]einer Mullisync Monitor	£349.00

PRINTERS

[illegible]ren Swift 9 Colour	£179.00
[illegible]ren Swift 240 Colour	£279.00
[illegible]ren Swift 200 Colour	£219.00
[illegible]500	£309.00
[illegible]500 Colour	£439.00
[illegible]550 Colour	£550.00

RAM

[illegible]1/2Mb	£14.95
[illegible]1/2 Mb With Clock	£19.95
[illegible]1Mb	£34.95
[illegible])Mb With Clock	£39.95
[illegible]) 2Mb (PCMCIA)	£119.00
[illegible]4Mb (PCMCIA)	£189.00
[illegible]Simms (Gvp)	£27.00
[illegible]Simms)Gvp)	£89.00
[illegible]2Mb - 8Mb	£129.00

SCANNERS

[illegible]er Mono	£95.00
[illegible]r Colour	£235.00
[illegible]Data Mono	£99.00
[illegible]ta Plus	£119.95
[illegible]ta OCR	£165.00
[illegible]er Mono	£89.00
[illegible]ner Scan Read	£129.00
[illegible]can Pro V3	£89.00
[illegible] GT 8000	£1199.00

HARD DRIVES

GASTEINER POWER

* External IDE HDD for Amiga A500/A500+/A1500/A2000
* Memory conveniently expaoded to 2/4/6/8Mb by using 1M X 4 Zips
* 100% compatible
* Easy Installation, Just Plug in and go
* Auto boot, Auto config and zero wait states

Controller for A500/A500+/A1500/ A2000	£99.00
Controller + 40Mb Hard Drive	£249.00
Controller + 65Mb Hard Drive	£279.00
Controller + 85Mb Hard Drive	£299.00
Controller + 120Mb Hard Drive	£329.00

GASTEINER POWER FOR A600/A1200

20MB + IDE Cable	£149.95
65Mb + IDE Cable	£199.00
85Mb + IDE Cable	£279.00
120Mb + IDE Cable	£299.00
Fitting for A600 or A1200	£29.95

BARE HARD DRIVES

IDE		SCSI	
40Mb	£99.00	50Mb	£199.00
65Mb	£199.00	85Mb	£279.00
85Mb	£219.00	120Mb	£299.00
120Mb	£249.00	210Mb	£319.00

ACCESSORIES

Power Supply A500/A500+/A600/A1200(High Voltage)	£34.95
Power Supply A1500/A2000	£69.95
Internal Drive for A500	£40.00
Internal Drive for A2000	£45.00
A500 Rom Switcher	£12.95
A600/A1200 Rom Switcher	£12.95
Auto sensing joystick/Mouseswitch Box	£12.95
Printer Cable	£6.00
Modem Cable	£6.00
RGB Cable	£6.00
IDE Cable for A600/A1200	£15.00
External Drive for A2000	£49.00
Blitz Amiga	£20.00
ICD Flicker Free Video 2	£199.00
Commodore 64 Power Supply	£19.00
10 Blank Disc	£7.00
3.5" External Drive	£50.00

MICE + TRACKBALL

AlfaData

Infra Red Mouse	£45.00
Mega Mouse	£10.95
Mega Mouse (Mat + Hulder)	£14.95
300 DPI Optical Mouse	£27.95
The Trackball	£29.95
Crystal Trackball	£34.95
Optical Pen Mouse	£35.95

Golden Image

GI-600	£13.95
Optical Mouse	£23.95
Brush Mouse	£19.95
New Golden Image 400 Dpi Mark 2 Mouse	£14.95

SOFTWARE

Word Processors/Publishing

Pen Pal V1.4	£49.95
Final Copy II V2.0	£99.95
Kindwords 3	£39.95
Wordworth V1.1	£109.95
Transwrite	£29.95
Prowrite 3.3	£79.95
Pagestream V2.2	£129.95
Professional Page V3.0	£129.95
Pagesetter II	£44.95

Softclips Clip Art

Classic Clip Art	£29.95
People Clip Art	£29.95
Collectors Clip Art	£29.95
Animal Clip Art	£29.95
Electric Thesaurus	£29.95

CAD & structuredrawing

Intro CAD Plus	£79.95
X-CAD 3000	£269.95
Professional Draw 3	£89.95

Animation and Graphics

Deluxe Paint 4	£64.95
Real 3D Professional Turbo	£249.95
Art Department professional V2	£144.95
DCTV Composite Video 24 Bit graphics System (PAL)	£379.95
Imagine 2.0	£189.95

Phone for access to our massive competitively priced range now!

[illegible]ducts advertised represent [illegible]mall sample of our instock [illegible]ge. A complete price list is [illegible] available on request.

DELIVERY CHARGES **UK MAINLAND (NOT HIGHLANDS)**

Small consumables & Software items — Despatched by post please check charges when ordering.
Other items, except lasers Offshore and Highlanders — Next day courier service, 10 per box
Please enquire

IN ADDITION WE OFFER THE FOLLOWING EXPRESS SERVICE

Saturday deliveries — Normal rate plus 15+ VAT per box
AM next day. — Normal rate plus 8+ VAT per box

E&O.E Price subject to changewithout notice. Goods subject to availability. Specifications subject to change without notice. All Trademarks Acknowledged.

SPRITES

When was the last time you looked at the graphics in a game and thought, 'I wish I could do that'? With AMOS you can!

Attention has to be paid to the backdrop. After all, these spiders wouldn't look quite the same if placed...

...on a racetrack! See what I mean?

The Amiga is capable of displaying eight hardware sprites on screen at once. AMOS is capable of displaying up to 64 computer sprites, all kept alive and healthy by the interpreter.

You might think that such a complicated business would require a complicated set of commands, but nothing could be further from the truth. AMOS Basic uses only seven commands to create and use sprites, and then hands over to AMAL to do the rest.

AMAL is the AMOS Animation Language, and is used to create smoothly animating and moving sprites which, once set, can be left to go about their business. To show you how easy it is, we're going to load a sprite and animate it. First, load up a sprite bank – either your own or the 'Spidy Abk' file on the coverdisk. (Go to direct mode to do this)

Now return to the Editor window and type the commands.

```
SPRITE 8, 200,100,1
DIRECT
```

Now run it. That was easy, and getting it moving is just as simple. Enter these lines

```
a$="Anim 8,(1,8)(2,8);"
a$=a$+"Loop:Move 320,0,100; Move -
320,0,100; Jump Loop"
AMAL 8,a$: AMAL On
```

Can you guess what the mysterious AMAL commands are? You'll have to wait until later to see if you're right.

SPRITE CONTROL

Here are all the AMOS Sprite commands, complete with syntax and examples.

SPRITE

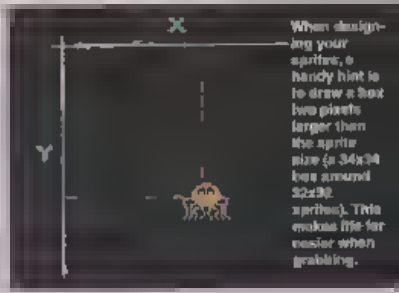
This command simply creates a sprite and displays it on screen. The instruction is followed by four variables, namely the index number of the sprite which can be anything between 0 and 63, the X and Y co-ordinates of the sprite and the sprite image which is to be taken from the sprite bank.

GET SPRITE PALETTE

It always happens. You have everything set, you load your sprites and when you display them, they look awful. The sprite bank holds the correct colours for the sprites, but unless stated otherwise the sprites take the palette from the current screen, which generally speaking is wrong. So, by sticking this command at the start of your program, you can correct this little problem.

SPRITE OFF

The Sprite Off command can be used in two ways. On its own, it turns off all sprite activity and removes all sprites from the screen. However, by adding a number to the end of the instruction you can specify a sprite to disable. Try the example in Table 1 (with a sprite bank in memory).



When designing your sprites, a handy hint is to draw a box two pixels larger than the sprite size (a 34x34 box around 32x32 sprites). This makes life far easier when grabbing.

TABLE 1

```
For a=1 to 8
sprite a,a*25,100+a,1
next a
locate 0,0 Print "Enter number of sprite to disable"
Input a
sprite off a
direct
```

SPRITE UPDATE

Sprite Update is an automatic process that tries to move all sprites during a vertical blank, creating smooth movement. However, if you have a lot of sprites on screen at once, it can't handle them and you end up with some noticeably jerky movements. Use the instruction Sprite Update Off to turn off the automatic process in situations like this, and Sprite Update On to switch it back on.

X SPRITE, Y SPRITE

X Sprite and Y Sprite are used to find the X and Y co-ordinates of a nominated sprite – useful when using AMAL Movement commands which don't tell a sprite to stop moving when it reaches the edge. The X Sprite and Y Sprite commands allow you to keep a check on all sprites using the syntax

```
variable=X Sprite (Sprite number)
```

GET SPRITE

The Get Sprite instruction does the same job as the sprite grabber in the Sprite Editor program, and allows you to take sprites directly from a screen image. If you know the co-ordinates of the images you want to grab, this is much faster. Load an IFF image to the current screen, and try these commands

```
Get Sprite 1,205,155 to 232,132
Get Sprite 2,150,100 to 205,155
```

Now display the sprites using the Sprite command, and see which areas you've grabbed.

TABLE 2

```
Sprite 8,100,100,1
a$="Anim 0,(1,8)(2,8);"
a$=a$+"Loop:Move 150,0,10; Move
-150,0,10; Jump Loop"
Amal 8,a$: Amal on 8
do
Print "Enter a word"
Input z$
Print z$
loop
```

AMAL SYNTAX

AMAL works using string variables – sets of instructions enclosed in quotation marks. Unlike standard AMOS commands, the program doesn't correct case or spacing, so you have to be very careful when entering your AMAL strings. If you entered the program above and got the error message 'Error in animation string', check that all the commands start with a capital letter, and that the semi-colons (;) are in the correct spaces.

Once you have created your AMAL string, it has to be assigned to an available sprite with the command AMAL (sprite number),(String variable name), and then switched on with the command AMAL On.

MOVE

The most basic of all AMAL commands is the Move instruction. As you might guess, it simply moves a sprite in a certain direction relative to its current position, at a set speed. Note: The co-ordinates you specify in the instruction tell the sprite how far to go, not which co-ordinate to move to. Co-ordinates of 100, 100 will move the sprite 100 pixels to the right and 100 pixels down from its present position. The third variable denotes the number of movement steps allocated. An instruction that moves the sprite 100 pixels using 50 steps will move the sprite two pixels at a time, giving quite smooth movement. Load the sprite bank from the disk and try the example in Table 3.

TABLE 3

```
Sprite 8,100,50,1
a$="Move 100,0,50;Move 0,100,50;Move -100,0,50;Move 0,-100,50;"
Amal 8,a$:Amal on
```

KEEP ON MOVING

AMAL has been developed for those people who really don't want to be bogged down with animating and moving sprites by hand, who would rather go without than track every single attack wave in a shoot 'em up. Basically, AMAL has been developed for everyone! It allows you to set movement and animation instructions to a sprite, and then go off and do other things. Load in the 'Spidy.Abk' sprite bank, and try the program in Table 2.

ANIM

Anim animates a sprite at a set speed through a pre-determined series of frames. To tell the program how to animate, pairs of numbers need to be entered into your string to tell the program which frames to display and for how long. Load the 'Spidy Abk' sprite bank and try the example in Table 4.

TABLE 4

```
Sprite 8,100,100,1
a$="Anim 0, (1,8)(2,8)(3,8)(4,8);"
Amal 8,a$:Amal on
```

Experiment with different speeds and orders to see how it works.

PLAY

The Play instruction tells the program to play an animation path defined in the AMAL editor. The command is followed by a number, which tells the interpreter which animation path from the AMAL memory bank to use. On the coverdisk is an AMAL bank called 'Fly Abk'. Load this from direct mode, and view the various flight patterns using the Play instruction.

AMAL ON, OFF, FREEZE

These three instructions cause all AMAL paths to start, stop, or pause until started again unless a specific sprite number is included. Enter the program in Table 5 and use the keys 1 to 4 to pause and restart the spiders.

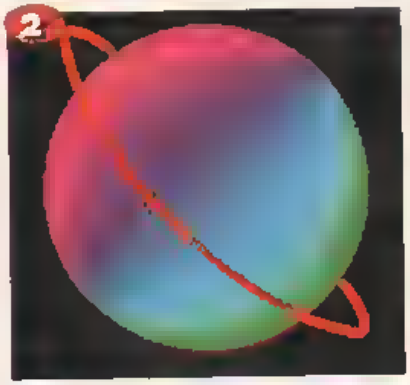
TABLE 5

```
Global a1,a2,a3,a4,x$, a$
Sprite 8,100,50,1
Sprite 9,140,50,1
Sprite 10,180,50,1
Sprite 11,240,50 1
a$="Anim 0, (1,8)(2,8);"
a$=a$+"Loop: Move 0,100 50; Move 0, -100,
50; Jump Loop;'
Amal 8, a$
Amal 9, a$
Amal 10, a$
Amal 11, a$
a1=0:a2=0,a3=0 a4=0
amal on
do
x=0
x$=Inkey$
If x$<>"" Than PSE
Loop
Procedure PSE
If x$="1" and a1=0 than Amal Freeze 8: a1=1:
Goto RTURN
If x$="1" and a1=1 than Amal On 8: a1=0
If x$="2" and a2=0 then Amal Freeze 9: a2=1.
Goto RTURN
If x$="2" and a2=1 then Amal On 9: a2=0
If x$="3" and a3=0 then Amal Freeze 10
a3=1: Goto RTURN
If x$="3" and a3=1 than Amal On 10: a3=0
If x$="4" and a4=0 then Amal Freeze 11:
a4=1: Goto RTURN
If x$="4" and a4=1 than Amal On 11: a4=0
RTURN:
End Proc
```

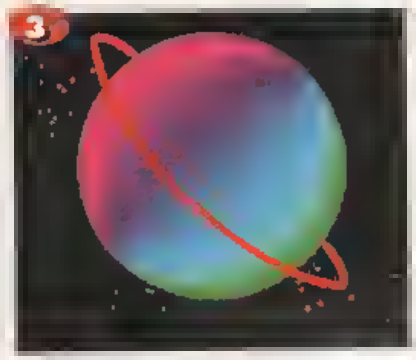
SCREENS

Load this screen into screen 1.

The Amiga's multiple screen modes make it a very versatile machine. However, Intuition – the software that controls the Workbench interface – is a pain to work with. AMOS screen control, on the other hand, makes light work of scrolling, flipping, animating, windows and a whole host of other functions.



Load this screen into screen 2.



Now use the dual playfield instruction to get something that looks like this! By scrolling screen 1 slightly faster than screen 2, you get a parallax effect.

Before you can display any graphics, text or other images, a screen has to be defined and displayed. When you first run it, AMOS has already done this for you, opening a 320x200 low-resolution screen, but what happens when you want more, such as a PAL display, HAM colours or high resolution? Simple – you create a new screen!

SCREEN OPEN, CLOSE

The Screen Open command defines a new screen and brings it to the front of the stack, making it the one currently displayed and written to. To open a screen, use the format:

Screen Open (Screen Number), (Width), (Height), (Colours), (Resolution)

So, to open a PAL, low-resolution screen with 64 colours, you would use the instruction

Screen Open 1, 320, 256, 64, Lowres

To open the same size screen in high resolution mode with 16 colours you would enter

Screen Open 1, 320, 256, 16, Hires

To close any screen, use the Screen Close instruction. This can be followed by a number, which denotes the screen to close, or closes the current window if left without.

SCREEN DISPLAY

With the Screen Display instruction, you can position your screen wherever you like on the monitor display, letting you create interesting 'bouncing screen' demos. The command is followed by five variables, which mark the screen number, the x position, the y position, the width of screen shown and height of screen shown respectively. See Table 1.

TABLE 1

```
Screen Open 1, 320,200,32,Lowres
For c=1 to 100
x1=rnd(300):y1=rnd(200):Ink rnd(32)
Bar x1,y1 to x1+50,y1+50
Next c
For c=90 to 150
 Screen display 1, C,,,,
 Wait Vbl
Next c
```

SCREEN OFFSET

The Screen Offset instruction lets you do all sorts of clever scrolling. It works by displaying the current screen from a specific point – but not necessarily the top left corner. This instruction works best if you have an extra large screen, and can be used to great effect. Try the routine in Table 2.

See how easy it is to smoothly scroll a screen? In case you aren't sure, the program is displaying the screen from X-position 'X', and reads the joystick in an endless loop. As the joystick is moved, 'X' is increased or decreased, and the screen is redisplayed.

Once you've got your screens up and running, there's a lot you can do with them without actually doing much at all. These screen effects contain most of the features and functions used in commercial games, but with none of the fuss.

DUAL PLAYFIELD

Parallax scrolling can really add something to your games, and the easiest way to create it is to use the Dual Playfield (screen one), (screen

SCREENS

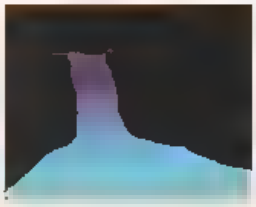
TABLE 2

```
Screen Open 1, 960,200,16,Lowres
For c=1 to 200
 X1=rnd(960):y1=rnd(200).Ink rnd (15)
 Bar x1,y1 to x1+rnd(50)+1,
y1+rnd(50)+1
Next C
x=0
Do
 Screen offset 1,x,0
 If Jleft(1) and x>1 then x=x-1
 If Jright(1) and x<640 then x=x+1
 If Fire(1) then Direct
 Wait Vbl
Loop
```

two) instruction. This takes two previously defined screens of the same resolution and overlays them, using the transparent colour (generally 0) to see through the top screen. The first screen is usually the one on top, but you can switch them around using the Dual Priority instruction. See the example given in Table 3

TABLE 3

```
Screen Open 1, 640,200,16, Lowres
Screen Open 2, 640,200,16, Lowres
Screen 1
For A=1 to 501 Step 50
 Bar a,0 to a+25, 200
Next A
Screen 2
For a=1 to 601 Step 25
 Ink Rnd(16)
 Bar a,0 to a+10,20: Bar a,180 to
a+10,200
Next A
Dual Playfield 2,1
x1=0:x2=0
Do
 Screen Offset 1,x1,0
 Screen Offset 2,x2,0
 If Jright(1) and x1<250 and x2<500
then x1=x1+1:x2=x2+2
 If Jleft(1) and x1>0 and x2>0 then
x1=x1-1:x2=x2-2
 If fire(1) then Direct
 Wait Vbl
Loop
```



Using the Shift Up instruction, animate this waterfall. Why not try some of your own!

SCREEN COPY

The Screen Copy command is the easiest way to duplicate an area of a screen and transfer it to another screen, or another area of the same screen. The format used is

Screen Copy (Screen Number), X1,Y1,X2,Y2 To (Screen Number), x3,y3

X1,Y1,X2 Y2 describe the rectangular area of the screen to be copied, and X3,Y3 mark the position where the top, left corner of the block will be pasted. Load the 'Copydemo' file from the coverdisk, go to direct mode, and try these examples.

```
Screen Copy 1,0,0,100,100 to 1,101,0
Screen Copy 1,0,0,320,100 to 1,0,101
```

APPEAR

This instruction lets you move smoothly between two pictures in a variety of ways — perfect for clearing the intro screen of a game! The instruction works by first identifying the source and destination screens, and then the effect, which can vary from one to the total number of pixels in your screen. Try the example in Table 4

TABLE 4

```
Screen open 1, 320,256, 32,Lowres
Load IFF "(Your screen)",1
Screen open 2, 320,256,32,Lowres
Load IFF "(Your Screen)",2
Appear 2 to 1,81920
Wait 200
Direct
```

FADE

The Fade command can be used in a variety of ways. In its most basic use, it fades all the colour registers to 0 (black) at a set speed, as in:

```
Fade 15
```

Or you can use it to change the colour registers to a new palette, as in:

```
Fade 15, $1,$2,$3,$4
```

Finally you can Fade the colours to a palette taken from another screen. Load two IFF files, and enter

```
Fade 15 to 1
```

If you have screen 2 displayed, this will change the palette to that of screen 1

SHIFT UP, DOWN

Colour cycling can be used to great effect, as anyone who has ever messed about with *Deluxe Paint* will tell you. Shift Up moves the colours in a certain range up a step at a time through that range, and Shift Down does the opposite. The last number in the instruction tells the interpreter what to do with the end colour in the range. Try this program:

```
Load Ill "Waterfall",1
Shift Up 10,6,10,1
Direct
```

See how impressive it can be?

PAINT THE WHOLE WORLD...

Copper rainbows are commonly used to create complex colour backdrops to games, allowing you to have far more colours on screen than you have in your palette. The instruction is laid out like this

Set Rainbow number, colour, length, red, green, blue

The number of your rainbow can be between 0 and 4. Colour is the colour index the rainbow will be based on. The length is the size of the table used to store your colour, ranging between 16 and 65500. The Red, Green and Blue indexes tell the program how to alter the basic colour index. The information for these is held in brackets, using the format (Number Of Lines, Amount to be added in a single step, Number of times to repeat the operation). See table 5.

TABLE 5

```
Set Rainbow
0,1,64,"(8,2,8)","(8,1,8)","
Rainbow 0,56,1,255
Wait Key
```

Notice how the Rainbow instruction is needed to display your set Rainbow. The syntax for this instruction is:

RAINBOW Number, first colour, vertical position, height.

SOUND

How good would your favourite game be without sound effects or music? Think about it – the sound really sets the atmosphere so you'd best get familiar with AMOS's set of sonic commands.

There are essentially two forms of sound in AMOS – samples and music. Each of these are held in their own designated memory banks and can be played across any sound channel. Samples are exactly that – raw sound that can be played at a requested rate. Music, however, is AMOS's version of a tracker module. All sounds and patterns are saved as one block, and accessed using a single command. Unfortunately, AMOS can only read AMOS music files, so you can't play your favourite tracker modules directly, but you can covert them to AMOS music files using a handy utility (see panel).

BELL, BOOM, SHOOT

AMOS has three sounds in memory at all times – a bell, a gunshot and an explosion effect. These are played using the commands Bell, Shoot and Boom respectively. Try the example in Table 1.

TABLE 1

```
For A=1 to 5
  Bell: Wait 5
Next A
For A=1 to 50
  Shoot: Wait 5
Next A
Boom: Wait 5: Boom
```

OK, so they may not be the most incredible effects that you have ever heard, but they'll certainly do until you start bringing in your own sounds.

SAM PLAY

Provided you have a sample bank in memory, you can play any of the sounds within it with the Sam Play command. The command is followed by three variables, the first is the number of the sample to play, the second is the sound channel it is to be played through (0 to 3), and the third is the playback rate.

SAM BANK

It's possible to hold more than one sample bank in memory at a time, and this command switches between them. To use it, simply type the command, followed by the number of the bank you want to switch to. If you aren't sure of the bank numbers, go to direct mode by pressing escape and type Listbank.

SAM LOOP

This command turns all samples into looping ones. To enable it, type SAM LOOP ON. To disable it again, type SAM LOOP OFF.

MUSIC

To play AMOS music files, you merely need to type the word Music followed by the number of the piece you want to hear. A music file can contain numerous pieces of music – one for each level of your game if you want – so including the number is vital. Without it, the command will play the first piece of music it comes across.

MUSIC OFF

Stops all music pronto. If you have more than one piece playing, and you only want to stop one track, then use the Music Stop command.

TEMPO

The Tempo command is used to alter the speed of any piece of AMOS music. The command is followed by a number which dictates the new tempo – the higher the number, the faster the music is played.

MVOLUME

Mvolume is short for Music Volume, and that's precisely what it is used to set. Ranging from 0 to 63, the command changes the volume of the entire piece, not just single tracks, but used with a loop can create some useful music and sound effects. Load the demo tune (spidy.abk) on this month's coverdisk and try the listing in Table 2.

TABLE 2

```
Load Iff "Title",1
Music
Wait Key
Fade 15
For a=63 to 0 step -2
Mvolume A
wait 5
next a
```

Professional looking, isn't it? That is exactly how easy it is to combine sound and graphics for stunning looking presentations, opening your AMOS world to more than just games.

HOW TO MAKE A SAMPLE BANK

To make a sample bank, you need to load the 'Sample Bank Maker' program on your AMOS Program disk and run it. You'll be shown a black screen with a menu bar. Using the right mouse button, select the 'Load Sample' option and a file requester appears.

Insert your disk of samples, and select the first one you want to include in the bank. The program will ask you for the sampling rate and then store it in memory. The sample will now be loaded and listed at the top of the screen. Repeat the process as many times as memory allows (watch the 'Memory Spare' indicator). Now just select the 'Save Bank' option from the menu, and the program does the rest.

HOW TO USE A TRACKER MODULE

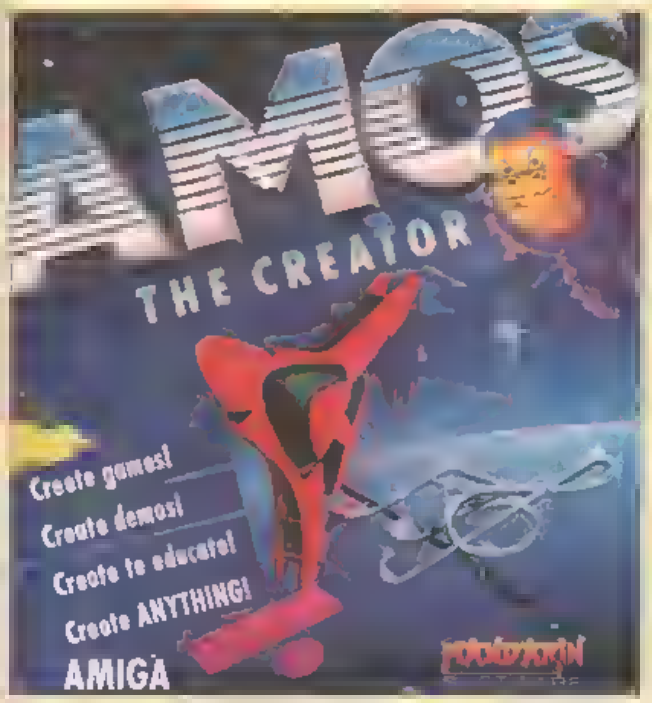
Before AMOS can play a tracker module (Noisetracker, Soundtracker etc), it needs to be converted to a format which AMOS understands – AMOS Music. This couldn't be easier if it was done for you. On your AMOS Program disk, you'll find a file called 'Soundtracker 2.1'.

Load this and run it, and a file requester will ask you for your tracker module. Insert the disk the module on it, and select it. The rest is done for you – all you have to do is specify a file name. To load the new file, go to direct mode and type "Load "(Filename).Abk", and the file will automatically be loaded into memory.

SOUND

SPECIAL OFFER

AMOS AND AMOS COMPILER DOCUMENTATION SET



Get the most from your free AMOS and Compiler with the official user documentation.

You've marvelled at the power and speed of AMOS and AMOS Compiler. You've tinkered with the demo programs, and maybe even created a few little routines of your own, but what now? If you really want to get the best from this stunning package there's no substitute for the official instruction manuals.

For starters, the AMOS manual explains in simple terms how the system works. As well as this, every command is listed in detail with clear examples and descriptions of each to get you up and running within minutes. Extensive technical appendices are also included for detailed information on the more obscure points.

Once you've got to grips with the basic AMOS language, you'll probably want to get things running even faster with the Compiler. The Compiler is available fully packaged, giving you a home for your disks and full instructions in one hit!

To complete your AMOS set, fill in the coupon below (or telephone your order on 0625 859333 quoting reference CU Amiga) indicating whether you require the AMOS manual, the Compiler manual and box, or both. The AMOS manual and Compiler set are each priced at £14.99. Cheques should be made payable to Europress Software Ltd. Alternatively, quote your Access/Visa card number, and the amount will be debited from your account.

NAME.....

ADDRESS.....

.....

.....

POSTCODE.....

PLEASE SEND ME:

❑ 1 AMOS manual @ £14.99

❑ 1 AMOS Compiler manual with box @ £14.99

❑ 1 each of AMOS manual and Compiler manual with box @ £24.99

❑ I enclose a cheque for £ , made payable to Europress Software Ltd.

Access/Visa card no.:.....

Expiry.....

Date:.....

Please supply credit card holder's address if different from address above.

Signature:.....

Send to: Europress Software, Europa House, Adlington Park, Macclesfield, SK10 4NP. Please allow 14 days for delivery.

AMOS COMPILER

You know already that you're fortunate enough to have the AMOS Compiler thrown in with your free copy of AMOS, but do you know how to get the best out of it?

For your Amiga to run an AMOS program, it has to run it first through the AMOS interpreter, which converts it into machine code, and then into the processor. This procedure takes time - which is where pure machine code programs have the edge. Or do they?

The AMOS Compiler is a handy accessory that takes your (sometimes) plodding AMOS files and converts them into pure machine code. The practical upshots of this are (a) it runs directly from disk, with no need to load the AMOS program and (b) with all conversion already done, the programs are vastly accelerated. Typically, compiled programs run at twice the speed of their BASIC counterparts.

How do you take advantage of this fabulous aid in your quest to get a game onto the shelves? The simplest way is to load the Compiler program from your main AMOS disk and compile from there, but there are other ways. The easiest is to compile from direct mode. Press escape to enter direct mode, and enter the compile command using this syntax:

Compile "(Program name) -(disks) -(type)"

The (disks) and (type) refer to the way the program is compiled and the type of file created. The complete list of settings is:

DISKS

- D00: Compiles from Ram Disk to Ram Disk. The fastest way to compile.
- D01: Compiles from Ram Disk to floppy disk.
- D10: Compiles from disk to RAM. It's fast, but it uses a lot of memory.
- D11: Compiles from floppy to floppy. Very slow, but only holds 70K of memory.

TYPES

- T0: Creates a Workbench friendly, stand alone file complete with icon.
- T1: Creates a CLI-friendly program, executable from the CLI.
- T2: Not the film, a CLI program that can run in the background using Amiga multi-tasking.
- T3: Creates a compiled AMOS program that has to be run from within AMOS.

So, to compile a program completely in RAM that can run as a CLI multitasking program you would enter:

Compile "(Program name) -D00 -T2"

After that, just follow the on-screen prompts.

OTHER OPTIONS

There are a couple of other lines you can add to your Compile command which give you more control over how the program will run when loaded independently. The first sets the default opening screen.

You'll find that AMOS compiled programs automatically open Screen 0 on loading, before running your program which can cause a nasty flash. To get rid of this, use the extender -S0 in your instruction.

You can also choose to keep the Workbench or CLI screen intact while your AMOS program sets itself up if you wish to, which will have the effect of making everything look far more professional.

To keep the Workbench screen up, use the extender -W1. Remember to put the line 'AMOS To Front' when your program is ready to display itself. Try the example in Table 1 on the left, compiling it as:

Compile "Test.AMOS -D01 -T2 -S0 -W1"

TABLE 1

```
Screen Open 0,320 250,64,Lowres
For C=0 To 100
  Ink
Rnd(64),X1=Rnd(320):X2=Rnd(320):Y1=Rnd(200):
Y2=Rnd(200)  Bar (x1,y1) to (x2,y2)
Next C
AMOS To Front
Wait Key
```

HOW TO COMPILE

For those who don't really feel like messing about with extenders and CLI-style commands, the Compiler.AMOS program on your AMOS program disk provides a useful alternative. Without any programming knowledge, you can compile your programs into fast machine code files faster than it takes to read this box! Here, in three easy stages, is the hassle-free compiling experience.

Load the compiler from the AMOS Program disk using the 'Load Others' option from the menu bar, and then click on 'Run Others'. Select the Compiler from the file selector and you'll be greeted with this menu screen. Here you select how the file is compiled, and what hardware is used. Along the top of the screen you'll see three icons. These represent the From, To and Type selectors. Click on each a couple of times to see how you can change the from and to between Ram and floppy disk. The Type box allows you to choose a WB compatible file, a CLI multitasking compatible file or an AMOS file. Choose the set-up that suits you, and click on the 'Compile' button.

The first of two file selectors appears. Select the AMOS file you want to compile, click on 'OK', and type the name you want the compiled file to be saved under. This file doesn't need to have an AMOS extender.

Now all you can do is [illegible] as the yellow bar inches up. After a few seconds, you'll be informed that all is done, and you can then load your compiled file and marvel at the speed.

COMPILER

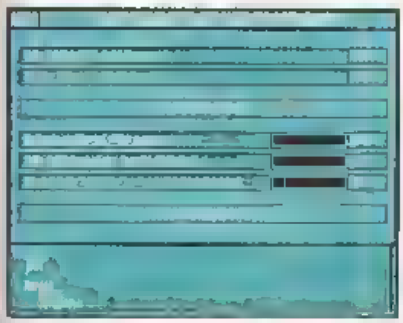
WHERE TO NOW?

You think you've seen all that AMOS has to offer? You ain't seen nothing yet! This is just the beginning – your first steps into an exciting new world. Just check out what AMOS has to offer you!

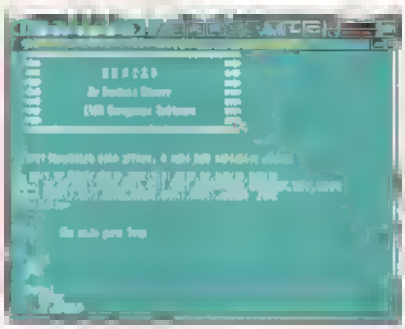
AMOS 3D lets you create anything from a business demonstration to a flight sim, using 30 new commands.

UPDATES

AMOS 1.35 is far different from the original version. Europress have made a point of sporadically releasing update disks for the system, comprising of new commands, friendlier accessories and more programming power than François Lionet ever imagined. The best thing about them, though, is that they are free! When one is available, it is instantly released to all PD libraries, not just the AMOS PSL, as well as on bulletin boards and available direct from Europress. At the moment, we're up to v1.35 – A1200 compatibility, but already we've seen improvements such as sprite flipping, full control over multi-tasking and an AMOS assembler! Roll on version 1.36!



These update disks can be found in any PD library, making your copy of AMOS more and more powerful



AMOS Pro is the big brother to AMOS, giving you over 750 commands and a whole host of new features. Well worth looking into

AMOS PRO

If the regular updates aren't enough for you, then why not fork out for a copy of *AMOS Pro* – this is not so much a game creator, more of a product development kit. With over 750 commands, a full debugging suite, marvellous WIMP driven user interface, and with a new update disk, which is compatible with AMOS 3D and the AMOS Compiler, you can't go far wrong.

Price £69.99 From Europress Software Tel 0625 859333

AMOS 3D

Any programmer will tell you that working with 3D polygon graphics can be a nightmare. Any programmer who hasn't used *AMOS 3D* that is. This extension to your AMOS interpreter lets you create and manipulate 3D objects as simply as moving a sprite, and that isn't all. The 3D Object Modeller lets you build objects in a way that *3D Construction Kit* could only dream of, allowing you to texture map surface detail onto the polygons, and then load them into AMOS and shift them around any way you like using 3D new commands. In BASIC, the graphics are fast enough, but compile them, and you've got speeds to rival commercial software!

Price £34.99 From Europress Software Tel 0625 859333

MAKE SOME MONEY!

Europress used to have a rule that any commercial software written in AMOS had to credit it, as well as display the AMOS logo within the game and on the packaging. As a result a lot of commercial publishers simply wouldn't take software written in AMOS. Not any more. Now, no mention need be made of your back door into quality software writing. All Europress ask is that you notify them of the release beforehand and send them a copy of the finished game when it's released. Europress Software reserve the right to release the information after two months, shellfish.

AMOS

PUBLIC DOMAIN

NOW ALSO AVAILABLE FROM HARGWARE IN AUSTRALIA

[illegible] CU, 11 YORK PLACE, NR BRANDON HILL, HOTWELLS, BRISTOL BS1 5UT

Strictly pd – guaranteed to make you merry this Christmas!

NEW GAMES

AS1 1 Tees Rollerpede Avatra +
AS1 2 Four classic games Get it! +
AS1 A Megaball and Drip Superb +
AS1 4 Frogger and more games +
AS1 5 Raid, Go Looly and Retaliator +
AS1 6 PacMan and four more
AS1 7 Moonbats Trix and Mightworks
AS1 8 Airace2 Hball and more +
AS1 9 Battlepong and Blizzard +
AS1 10 Escape Pipeline and Pickout +
AS1 11 Dad Connex Paverai+Tappin +
AS1 12 Llamatron, Car Powerpong
AS1 13 Twintris + many more games
AS1 14 ChinaChallenge2 A Amiga columns +
AS1 15 Battlecars Lattix and Mambamove!
AS1 16 Wastelands Mission X and PacMan +
AS1 17 Tennical, Defenda Jetman Bugblaster +
AS1 18 Henry in Panic Skyflyer Omega race
AS1 19 Growth FranticFreddy and more!
AS1 24 Revenge DuxCreak 3d maze +
AS1 25 Another mega collection – Buy it! +
AS1 26 SuperPacMan 92 Smash TV Ashido +
AS1 27 Addictive card games – Very good! +
AS1 28 Doody/DrMario/Invaders2/Madbomber2 +
AS1 29 DOG/Rome Neva/It-Bal/Atax/Queenx +
AS1 30 Bounce and Blast, Total Fire etc +
ENT + 1 Klondyke Hball and more +
ENT + 2 Zerg BounceNbals Froggie +
ENT + 3 SpaceWars Squamble and more +
ENT + 4 Zoo Hoop Yelp and more +

CLIP ART

- The clip art in this collection is v.high quality
- Huge bitmaps many greater than an Amiga Hires screen
- For use in Dpaint/DTP packages
- Each disk autoboots as a slideshow allowing you to view the entire bitmap by moving the mouse

ART01 Weddings clip art (2 disks) +
ART02 Houses – pictures of mansions (2 disks) +
ART03 Religious – full of holy clip art (3 disks) +
ART04 WW1 Aircraft (1 disks) +
ART05 Men 2 disks full of men at work +
ART06 Women 2 disks full of women! +
ART07 Kids clip art (2 disks) +
ART08 Business – office clip art (2 disks) +
ART09 Office – more of the same +
ART10 Students clip art +
ART11 Saletime – useful stuff inc logos +
ART12 Xmas – a lot of Xmas cheer (3 disks) +
ART13 Punch – cartoons etc (3 disks) +
ART14 Animals – v high quality pics +
ART1A Cats – 12 feline pics +
ART15 Silhouettes – 15 pics +
ART17 Schools (2 disks) +
ART18 Babies +
ART19 Sports – (2 disks) +
ART20 Olympic – more sporting stuff (2 disks) +
ART21 Medical clip art (2 disks) +
ART22 Sealife – In deep clip art +
ART23 Decorative months – pic for every month +
ART24 Wacky comedy clip art +
ART25 Holidays +
ART26 Banners (2 disks) +
ART27 Art Nouveau – floral art +
ART28 Suburbes +
ART29 Food – culinary clips (2 disks) +
ART30 Colouring Book – Colour in Dpaint +
ART31 Borders – for use in DTP (2 disks) +
ART32 Mac Clip – quality macart (2 disks) +
ART33 Hornback – scenic clip art +
ART34 Teddy Bears (2 disks) +
ART35 Vanity Fair – fashion clips (2 disks) +
ART36 Halloween clip art +
ART37 Showtime – dancing,singing etc +
ART38 Vegetation clip art +
ART39 Floral – 14 pics +
ART40 Mythical – stars and zodiacs +
ART41 Dog woodcuts +
ART42 Cat woodcuts +
ART43 African woodcuts – more animals +
ART44 American woodcuts +
ART45 Arctic woodcuts +
ART46 Farm woodcuts +
ART47 Mixed animal woodcuts +
ART48 Birds woodcuts +
ART49 More animals +
ART50 Animal lineart +
ART51 Botanical line art +
ART52 Family Scenes (2 disks) +
ART53 Golf humour (2 disks) +
ART54 Space and transport +
ART5A Softscene2 – rural clip art +
ART56 Sports 3 +
ART57 Navy – lots of navy type pics +
ART58 Travel (2 disks) +
ART59 Border and alphabet clips +
ART60 Accents and flashes +

We now have FRED FISH 1-750

LATEST UTILITIES

U361 The Final Frontier Vol 3 (4 disks) latest edition of this brilliant Star Trek mag +
UU027 Filofax – reminds you of important appointments +
UU028 Bible (4 disks) The complete King James Bible
UU029 Tarot It read your own stars
U364 Cyclops v1 0 – art creation package – create cinemas etc +
U527 Professional D-Copy v3 9 – as good as most commercial copiers
U524 IntroMaker Kit – make own intro screens with IFF pics digitised samples and scrolled messages +
U528 Multiplayer – claims to be the most powerful versatile music player for the Amiga +
U517 Keirens Font Editors/Boot Menu Maker
U325 Label Maker – make your own labels
UU030 Understanding Amos – tutorial +
U506 175 Utilities – the most on one disk
U533 Thrallbound – contains spellcheck v1 3 and Examiner v1 0 which help you to learn and tests you on foreign words Also contains good text adventure game +
U534 The main event – calender & alarm clock +
U531 Powerbase v3.2 – powerful and user-friendly database +
U535 Wordpower v1 3 – spell checking crosswords anagram solving +
U514 PC Emulator – runs most pc business software +
U365 Pixel (13 disks) – 13 disks crammed with programs etc. Entirely for A500/A600 users +
UU032 TDH Vol 1 – Amos users magazine +
U306 Strictly PD essential collection (4 disks) packed with utilities games music creators and printer utilities +

USEFUL UTILITIES

UU001 A GRAPH – Very easy to use – make bar + line graphs + pie charts +
UU002 CLI TUTORIAL – Learn all about the CLI Given 70% rating in this magazine +
UU003 MESSYSID V2 – Reads and writes PC disks Given magazine rating of 85% +
UU004 POOLS PREDICTOR – Increase your chances of becoming a millionaire +
UU039 DISK OPTIMISER – Brilliantly simple! Any disk loads up to 15 times faster Mag rating of 90% +
UU030 A 500 PLUS EMULATOR – Emulate the Amiga Plus on your 1 3 Amiga (IMG Req).
UU005 FREECOPY V1 1 – Removes password protection to allow copying +
UU009 CURSOR V1 0 – Compiler for Amiga Basic programmes +
UU008 BROWSER II – An alternative to SID Some may find it easier and better +
UU016 PC TASK V1 04 – Powerful multi tasking PC emulator supports CGA graphics, MS DOS floppies and even your hard drive +
UU017 NIB V2 0 – Removes protection and copies Very powerful +
UU012 GATORS GRAPHIC GALLERY – Examples and step by step tutorials on how to create professional logos like ones found in demos +
UU01A FISH TANK SIM – Simulates an aquarium +
UU014 ANALYTICALE SPREAD (2 DISKS) – This is the best spreadsheet program to date on the Amiga +
UU01A AMIGA PUNT PROGRAM – Predict which horse will win from past form +
UU019 JR COMM V1 02 – Latest version of this useful modem program +
UU017 BEATPIX POTTER CLIP-ART – An excellent collection of clip art for any DTP program or D-Paint +
UU016 PLOTTING AND GRAPHICS (2 DISKS) – Contains a full featured plotting prog, and a computer aided drafting prog +
UU019 DB – A database with up to 50 fields max of about 1.2 million records +
UU030 HOME MANAGER – A great all in one address book with an inventory database + to do list +
UU021 ASTRO PRO ASTROLOGY – The best astrology program on the Amiga by far +
UU002 MULTIDOS V1 12 – After installation your Amiga drives can read 1MB disks +
UU024 TEXT PLUS V4 0(E) – Latest update of this excellent word processor program Now compatible with TeX the professional typesetting prog Found in the Fish Collection +
UU025 AMOS LESSON I – Find out how to get the most out of AMOS Had good review +
UU026 EDWORD 2.2 – Best text editor around +
UU027 MEGACD/OUR V1 9 – A program which transforms a b/w screen into full colour +

We now have T-BAG 1-61

GAMES & EDUCATION

75p

N001 [illegible] CONCERTS (2 DISKS) – Story books about astronomy and dinosaurs +
N002 AMERICAN FOOTBALL COACH – Play the role of an American Football coach +
N003 BATTLEMENTS – Similar to Hunchback on the Spectrum and C64 +
N004 OTHELLO – The best PD version of this classic board game +
N005 CRAZY SUE – The best PD platform
N006 SUPERLEAGUE MANAGER – Football management game +
N007 METRO – Mmm aim silly +
N008 SMASH TV THE RIP OFF – Great stuff +
N009 HENRY IN PANIC – Jet set Willy clone +
N010 MISTER + MISAIE – Amusing and cute platform/arcade adventure +
N011 DUNGEONS OF AVALON – Compares to likes of Dungeon Master Stunning graphics and atmospheric sound effects. +
N012 CALORIEBASE – Received 88% in CU Amiga. Keep a daily database of your calorie intake. Useful for dieters +
N013 BACK TALK – Discusses common lower back problems, the causes and treatments given Uses detailed pics/anims +
N014 TRAINING LOG – for fitness training +
N01A SPANISH,FRENCH GERMAN,SPANISH TUTORS – 4 great tutors +
N01A A VISIT TO THE RED PLANET – Guided tour of Mars fascinating +
G396 FORMULA ONE CHALLENGE – 4 people can play in this excellent strategy match +
G397 TANK ATTACK – Attempt to storm HQ avoiding enemy tanks – good shoot em up +
G398 TOP SECRET – great text adventure with cute graphics, given 100% in mag review +
G399 TOMCAT – fly your F15 against enemy planes, ships and gun towers +
G400 FIGHTER PILOT – superb WW1 game +
G380 CARD BOARD A ARCADE GAME – 13 great games on one disk +
G381 PD PUZZLES – inc. puzzle + puzzle mania

DEMOS & MUSIC

D48X FRACTALE – excellent history of fractals
D483 ALACATRAZ ODYSSEY (4 disks) – probably the year's best demo
D487 WWF Pics – the clearest in-res pics I've seen
D499 RAUNCHY SLIDES – pics of pretty girls
D507 SCHWARZTOONS (3 disks) – packed with superb cartoon demos by Eric Schwarz
D199 SAM FOX SLIDESHOW
D128 MARIA WHITTAKER SLIDESHOW
D479 KATHY LLOYD SLIDESHOW
D01A MARIA WHITTAKER ANIMATION
D170 GIRLS OF SPORTS ILLUSTRATED
D500 DIGITAL DAMSELS
M374 HARDCORE III – pumping rave music and graphics
M379 TECHNOMANIA A1 – wire this into your hi-fi [illegible]
M378 NEW WAVE – slip on your [illegible] and boogie down
M406 EXPRESSIONS 3 – 5 tracks with 27 minutes worth of brilliant house/rave music
M364 ALCATRAZ – more than music, highly original music and demos
M362 CHART TECHNO 1 – 25 minutes of altern-8 remixes
M365 BANGING RAVES – the 2nd coming, 11 bands in the great remix
M366 SUPERSYNTHS VOL 1A2 (1MB) – great graphics and music

75p

VIDEO PRODUCTION

V001 DESKTOP VIDEO PACK – Collection for video producers including Rolling Credits Slideshow Video Backgrounds. Special Effects Pattern Generator and more!!! +
V002 S-MOVIE – Smooth scrolling video titler +
V003 Turbotitler – Add subtitles to your videos. +
V004 TV & VIDEO GRAPHICS (6 DISKS) – Packed with background screens for your video productions +
V004 VIDEO PRODUCTION (2 DISKS) – Packed with video & Genlock utilities +
V006 VIDEO SCREENS 1 – Background pictures for video production +
V007 VIDEO & ANIM VIDEO DB – Keeps track of your video tape collection RTAP lets you run large Anims on small MEM machines +
V008 IMAGE LAB – Like a mini art dept. Tools on tap does fades colour bars & grey bars +
V009 VIDEO STILL STORE – Used to create over the shoulder Graphic Inserts like the 6 O'Clock news +
V0010 SHADES & FADES ANIMFADER – A utility to fade screens in and out +
V0011 ANDROIDS BACK VIDEO GROUNDS – Selection of B/C pictures +
V0012 HARLEQUIN VIDEO ART & FONT DISKS (3 DISKS) – After rave reviews in CU Amiga shot to No.1 in sales chart. Excellent. +

CREATIVE UTILITIES

CR001 AMIGAVENTURE & AP VSYS – Make your own text games with these progs +
CR002 PRINT STUDIO – Excellent for printing pictures or text +
CR003 ANIMATION STUDIO – A brilliant Anim creator for you to make your own animations +
CR004 VERTEX MODELLING – Allows you to create 3D objects without using the X, Y & Z views. Loads sculpt3D/4D & Turbo Silver +
CR005 MODELLING OBJECTS – Contains over 25 vector objects in Image format. Perfect for use with CR004 +
CR006 MAGNETIC PAGES V 1 30 – Create your own disk magazine. Received 10/10 in review by Amiga shopper mag +
CR007 STRATA V1 9 – Landscape generator which allows printing of them from any angle any position & any magnification +
CR008 AMOSBROT V1 1 (A) – The people has even more fractal types which include Logistic Equation. Coast Lines & Trees – received good reviews +
CR009 GRAPHIC PRODUCTION – Utilities collection inc scenery V1 0 Landscape Landbuild V3 2 Cloud 9. Genesis demo +
U24T FONTS/LOGOS – For use with D Paint +
U237 PROFESSIONAL DEMO MAKER – Create your own demo masterpieces +
U299 SLIDESHOW MAKER – Shows how to present slideshows in different ways +
U064 SPEECH TOY – Get your Amiga talking +
U062 VOICES – Add speech to your demos etc +
U012 MANDLEBROTS – The best Mandlebrot generator around +
U31A ULTIMATE ICONS – Includes Icon Lab. Icon Magic & Icon Maker – splendid disk +
U32T SUPA FONTS – For use with D-Paint etc +
U071 ELECTROCAD – Impressive package +
U075 MCAD – Great 3D Design Package +
U331 LAND BUILD 2 2 – Generate landscapes and view from any angle Amazing! +
U346 ULTRAPAINT – Fab paint package (+)
U364 SCREEN MOD – Customise the way programs and screens appear +
U309 SHADOW DEMO MAKER – 3 Brilliant demo making utilities +
U500 DEMO CREATOR PACK (5 DISKS) – Get the most out of your Amiga – Get creative
U481 DKB TRACE – Excellent Ray Trace prog

- Cheques/P.O.s payable to STRICTLY PD

- **Buy 30 or more disks for just 75p each**
- Over 21 disks ONLY 85p EACH
- Only 99p per disk when you order 11 or more*
- Orders of 10 or less pay £1 25 per disk
- Please add £1 to all UK orders for first class postage Orders from Europe please add 25p per disk and Rest of World add 50p per disk for extra postage costs.
- Catalogue disk available only £1 Reviews of over 1000 disks + loads more

STRICTLY pd

DEPT CU, 11 YORK PLACE, NR BRANDON HILL, HOTWELLS, BRISTOL BS1 5UT

- The complete Strictly PD library is now available in Australia. To order a catalogue please send a cheque or postal order for $2.00 to Hargware [illegible] AC 29 Woralu St, Woramanga, ACT 2611 Australia

TOTALLY AMOS

When you start using AMOS, you're doing more than just using a programming language. Before you know it, you could find yourself with a new circle of friends!

The *Totally AMOS* main menu. Options can be selected with the keyboard or by clicking on the numbers with the mouse

Tutorials like this one are all well and good, but what happens when you come across a problem that you just can't solve? This booklet doesn't cover everything AMOS has to offer, and unfortunately neither does the manual. Basically, beginners can get a really hard time of it, but where can they turn for help?

To the husband and wife team of Len and Anne Tucker, that's where. These two have offered strong support for AMOS right from the very start, with Anne heading up the AMOS PD Library and Len offering technical support, as well as writing educational software such as Europress's *Spelling Fair* and Jumping Bean's *Noddy's Playtime*. Eighteen months ago, they put together the first issue of *Totally AMOS*, the disk magazine for the beginner.

'We saw a need for some sort of set-up to help the complete novice,' Len explains. 'We looked around at the time, and couldn't find anything that was subject specific. Everything seemed to assume that people knew what a For Next loop was, or what a While Wend was. We set up *Totally AMOS* to help people who needed it. While to us, and we'll do a tutorial on it, that sort of thing. Another aim behind *Totally AMOS* was to set up connections between programmers and artists, artists and musicians and so on. Both things were what we saw was needed, and we tried to create this environment – something like a beginners' club. What we really want is for members to feed off each other's knowledge.'

For the record, the entire thing was Anne's idea, and consequently she does most of the work in terms of putting the magazine together. Len is mainly responsible for the magazine driver, which is being continually enhanced. But before I go any further, let's take a look at the product itself.

Totally AMOS works from an interactive menu and displays text pages and illustrations at your command – a cross between Multimedia and teletext in that sense. Everything is controlled from the mouse or numeric keys, so there's no confusion from the start. But that isn't going to sell it.

What will, though, is the editorial content. Broken into 10 main sections, each broken down further, the disk contains reviews of AMOS PD and AMOS support titles, comprehensive news and letters pages, a debating corner – where readers can slag each other and the editorial team off as much as they want – and, of course, the help pages.

Help comes in two different ways. The first is from a Question and Answer session, where the smallest and simplest problems are solved. Other help comes in the form of complete tutorials, covering all aspects of a problem. Subjects covered in tutorials in past issues include AMOS 3D, How To Get The Most Out Of The AMOS Compiler, AMAL and a guide to the AMOS commands not mentioned in the manual.

One other feature is a spotlight on leading AMOS programmers – those who have created the most impressive public domain software and routines. If there's no other reason to work hard at your coding, the promise of an interview feature must be enough to entice most to submit work.

At the moment, *Totally AMOS* sells around 150 copies, but that looks set to change thanks to a new distribution deal that will see the magazine on sale in Canada and the USA, and then Australia.

Totally AMOS costs £2.50 per issue, and back issues cost £3.00 each. If you subscribe, you become eligible for a 10 per cent discount on all disks from the AMOS PDL. If you want to try it out, there is a PD issue available from the AMOS PDL for £2.00. Whatever space on the disk isn't taken up with the magazine is filled with useful routines and programs, making it a serious bargain. For more information, contact Len or Anne on (0792) 588156.

AMOS IN ACTION

As well as running the AMOS PDL, writing *Totally AMOS* and various educational packages, Len and Anne are also responsible for a new AMOS book simply titled AMOS In Action. The book is essentially a guide on how to write an arcade game in AMOS Basic, taking you from the sort of set up you really need to write a game, right up to completing the project. A disk is included containing a complete version of the demonstration game in question – Marvin The Martian – and costs a mere £12.95. If you buy it from the AMOS PD Library, it costs an exact mere £10.35. For more information, call Len or Anne on (0792) 588156.

Just one of the many Help pages, to get you out of those sticky situations.

AMOS PUBLIC DOMAIN

So what exactly can be done with AMOS? One place to start looking is in the various public domain libraries, where dozens of disks containing AMOS programs and routines can be found.

CARD GAMES 2
game

Everyone has those moments when you want to do nothing other than sit down with a pack of cards, and deal a quick hand of Patience. Or at least David Lomas seems to think so – or he wouldn't have come up with *Card Games 2*, a collection of nine different Patience variants. Being on your own needn't be a chore any more!

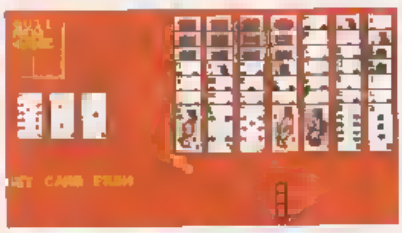
In case you're not the kind of person who enjoys spending long evenings alone, Patience is a card game for one. Generally it involves a number of card stacks, which have to be rearranged using a series of set rules to reach a certain position – four rows of ascending cards of individual suits, for example.

On the whole the game isn't particularly taxing – it depends on the luck of the deal more than anything else – but it does while away the time.

All nine games are accessed from a single menu screen, and to be honest there isn't a great deal of difference between them. Each game is displayed on the same blank backdrop with the same set of cards and an identical control method involving two clicks with the left mouse button – one to pick up a card and another to put it down again. The presentation isn't much, but it's such a great version that I haven't any time left to write this article!

Disk No: APD446. From: AMOS PDL, 1 Penmynydd Road, Swansea, SA5 7EH. Tel: 0792 588156. Price: £2.00. Compatibility: All machines. Memory: 512K

76%



For those moments when you have nothing to do – AMOS Patience!

FOOTBALL/SPEEDY READY
game

Football is possibly the most pointless management game around, and that's what's so great about it. Take something like *Tracksuit Manager*, remove all traces of management so that all you have left is the results screen, and you've got *Football*. It sounds like a strange idea, but you do find yourself clicking through the screens just to see who wins the league. No playability or gameplay, but fun.

Speedy Reedy, however, is playable. Playable and a lot of fun. In this *PacMan*-style maze game, the aim is to eat the power pills as they appear while staying out of the clutches of the evil ghost. It's all rather unfair, as the ghost can float through walls and you can't, but help is at hand. Collect a speed-up and you can race all over the shop without fear of being caught. Superb samples and music really make the game stand out – they just have to be heard to be believed.

Disk No: APD462. From: AMOS PDL, 1 Penmynydd Road, Swansea, SA5 7EH. Tel: 0792 588156. Price: £2.00. Compatibility: All machines. Memory: 512K

72%

It might look like PacMan, but that's where the resemblance ends with SpeedyReedy

U-TILITIES
utilities

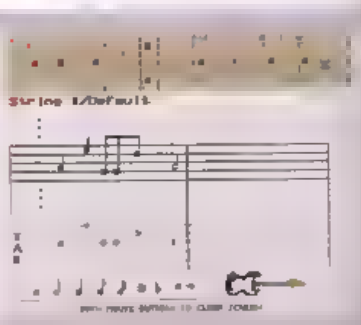
Just to show that AMOS isn't solely for making games, Tony Swanwick's dynamic duo of applications for outclass a lot of the more commercial PD utilities around. The first program, *U-File*, is a fully comprehensive file editor allowing you to load individual files from disk and tailor them to your own desires. If you would rather have your name than Mike Singleton's on the title screen of *Midwinter 2*, then this is the gizmo to do it with. To be used carefully.

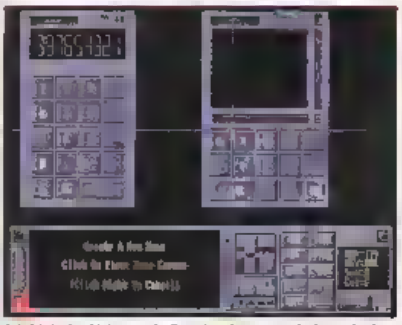
The other program, *U-Zone*, is an AMOS help application that lets you define screen zones for an IFF graphic. Anyone who has created a menu screen, and then spent hours trying to get the positioning of the buttons pixel perfect will know how much of a struggle it can be. This package is dedicated to you. Thanks to a few icons and some very well-written code, you'll be able to define every screen zone as easily as drawing a box on *DeluxePaint*. A must for all AMOS owners.

Disk no: APD454. From: AMOS PDL, 1 Penmynydd Road, Penlan, Swansea, SA5 7EH. Tel: 0792 586156 Price: £2.00. Compatibility: All machines. Memory: 512K.

91%

U-Zone is a godsend for people who like using buttons, but don't like making them!





A bit late for Valentine's Day, but fast nonetheless. A shot from *Digital Orgasm*.

DIGITAL ORGASM
demo

Despite the somewhat dubious title this demo, which comes from coders Cubic, is suitable for all the family and serves to prove that AMOS is just as good at creating light sourced vector objects and dot flags as most other programming utilities are

The nicest thing about it is that it's the latest in a long line of demos flying back and forth in a friendly competition between Cubic and coding rivals Fenalix, and they just get better and better Every time one team comes up with something new, the other has to reply by doing the same thing, only bigger better and faster

If you are a collector of megademos, then you probably won't find much to impress you here. After all machine code routines will always be faster than AMOS, but if you take into account that these routines are written in a version of BASIC, you can't help but be impressed

All in all, though, this isn't a very entertaining demo. I would have liked to have seen a lot more visual effects and a lot less text on the screen, or even some more entertaining text. Still, it shows what AMOS is capable of so far

Disk no: APD456. From: AMOS PDL, 1 Penmynydd Road, Penlan, Swansea, SA5 7EH. Tel: 0792 588156 Price: £2.00. Compatibility: All machines. Memory: 512K.

71%

Turbo Text is an excellent word processor – and a snip at £2.00!

TURBO TEXT
text editor

The mysteriously named Harbindar Ghag is responsible for this handy AMOS word processor. The screen layout is more or less the same as most others (ruler bar at the top of the screen, most options selected from a menu bar) but that's where most of the similarity ends.

All the usual options are included, such as loading and saving ASCII files (which makes this perfect for writing your AMOS routines on), and various formatting controls – which are, incidentally, perfectly arranged They do exactly what you would expect, unlike many PD word processors, which seem to have more than a few unpredictable results On top of these are a few options not normally seen For a start, you can set the word processor to read your text as you type Unfortunately it uses the Amige speech synthesis which everyone knows is about as decipherable as the old Spectrum Chrah Speech unit, but it works well in allowing you to keep your eyes off the screen if you should so desire You can also get the program to read the entire document back to you, which gives you an excellent way of looking over what you've written if you don't like reading your own work

Disk no: GPD145. From: AMOS PDL, 1 Penmynydd Road, Penlan, Swansea, SA5 7EH. Tel: 0792 588156 Price: £2.00. Compatibility: A500, A500+, A600. Memory: 512K.

92%

A must for any guitar player, Tab Master will solve those Tabulature blues and let you get back to playing them instead

TAB MASTER/HECTIC 2/DEAR DIARY
miscellaneous

There are three cool programs written by one David Meager included on this disk. He may be only 14 years of age, but he's already creating professional looking software! *Tab Master* is a must for any guitar owner, allowing you to enter musical notation on a stave, which is then converted into guitar tablature instantly and, if you want, is marked out on a fretboard No more messing around with mnemonics for me Next time I want to transcribe Mendelssohn, I'll just use this

Alongside it on the disk is *Hectic 2*, an interesting tile-based puzzle game which involves picking up numbered tiles to get the highest score possible Some tiles take points off your score, and some add to it

This might seem a little on the easy side but when you add to that the fact that the first player can only move the cursor horizontally on the board and the second player can only move it vertically, you realise that you've actually got yourself a real challenge

The diary program, which is simply titled *Dear Diary*, is really nothing to write home about I'm afraid, but when put on a disk with two great programs like these, you can't really complain.

Disk no: GPD180. From: AMOS PDL, 1 Penmynydd Road, Penlan, Swansea, SA5 7EH. Tel: 0792 588156 Price: £2.00. Compatibility: All machined. Memory: 512K

87%

PUBLIC DOMAIN

AMOS USER GROUPS

Being an AMOS user can boost your social life! This is the claim we make based on the sheer number of AMOS User groups there are in Europe alone. Here's a complete list of who to write to.

AMOS User Club UK
Aaron Fothergill
1 Lower Moor
Whiddon Valley
Barnstaple
North Devon
EX32 8NW

AMOS Programmer's Exchange
7 Majestic Road
Hatch Warren
Basingstoke
Hampshire
RG22 4XD

Klub AMOS France
BP 133
18003 Bourges Cedex,
France

Tom Poulsen
Danish AMOS Group
Stenmollen 28
2640 Hedehusene
Denmark

AMOS Club Nederland
Karkelnd 8a
5293 AB Gemonda (NB)
Holland

Belgium Club
Johan Francois
Wilganpark 7
9800 EEKLO
Belgium

AMOS Club USA
Mark H. Budzlazewski & Mark A. Shultz
PO Box 11434
Milw.
WIS 53211,
USA

AMOS NTSC Club
David Lazarek
516 E 11th Street
Michigan City
IN 46360,
USA

Aaron Wald
201-19 Tonnele Avenue
Jersey City
NJ 07306,
USA

Deutsche
Carsten Bernhard
Asternweg 4
6229 Walluf
Germany

Portugal
Eduardo David
Rua Nina Marquas Pereira N 9 2 - Esq
1500 Lisboa
Portugal

WIN A STACK OF DISKS

How do you fancy putting your new found knowledge to the test? How would you like to win 30 – yes THIRTY disks of your choice from the AMOS PD Library? Like the sound of that? Here's what you have to do. On the coverdisk is a program called 'Spidy AMOS'. This is a very basic program based on a Mac program called *Neko*. At the moment, all that you do is move a spider with the mouse pointer, trying to keep it out of its grasp while at the same time keeping it interested enough to chase. What we want you to do is soup it up.

Yes, you have the basic program, now see what you can do with it! Maybe the Spider should carry a rocket launcher? Maybe the screen should scroll? What do you think? You have complete freedom to do whatever you like. The best entry wins. It's as simple as that. So what are you waiting for! Get it together, and stick your entry on a disk and pop it off to us at CU with a covering letter explaining the changes made. Remember to mark your envelope 'Magnetic Media – Do Not Xray'. Send your disks to: I WANT ALL THAT LOVELY PD, CU Amiga, Priory Court, 30-32 Farringdon Lane, London EC1R 3AU. Closing date is 30th June. The editor's decision is final and no correspondence will be entered into. Employees of EMAP Images or the AMOS PD Library are not allowed to enter, although we can't figure out who from the AMOS PD Library would want to. After all, they've already got the disks, haven't they?

USER GROUPS

CUT, TICK & POST...... CUT, TICK & POST

TELEPHONE OR FAX HOTLINE (0702) 466933

PD SOFT (CU48) 1 BRYANT AVE, SOUTHEND-ON-SEA, ESSEX, SS1 2YD

EDUCATIONAL

FONT COLLECTIONS

GAMES SELECTION

ASSASSINS COLLECTION £49.99

ASSASSINS GAMES PACK

V704 AMOS COMPILER

V713 CHINA

V981 BENCH

V3022 LITTLE TRAVELLER v1.1

V3023 QUIZ

1778 ROULETTE ROYAL

1779 SOCCER CARD

2808 LADY BUG

2114 BULL RUN

2842 CRAZY SUE

2854 SPACE INVADERS

1738 AIR TRAFFIC CONTROL

Update 9 OUT NOW, Phone or send 1 SAE for a Free copy

HISTORY OF HOUSE MUSIC

ACCOUSTIC SILENCE

ORDERING BY POST Please tick the boxes in the right of the disk number... After selecting your order please fill in your Personal details on the order form provided. CUT OUT or Photocopy this ... Hand written orders are also accepted. Cheques & Postal orders should be made payable to PD-SOFT.

TELEPHONE ORDERS Credit Card orders can be accepted ...

COLLECTING DISKS You can now collect any disks ... Southend On Sea ... Orders are quicker if you ring in advance.

EEC ORDERS When ordering from within Europe ... Eurocheques are accepted ...

INTERNATIONAL ORDERS All orders are sent by Air Mail ... Cover any postage costs ...

NAME	
ADDRESS	
POSTCODE	A/c No	TELEPHONE
AMIGA PLUS OWNERS (A) IT WORKS OK	CREDIT CARD NUMBER (MASTERCARD or VISA)	EXPIRY DATE	TOTAL PRICE	TOTAL PD DISKS

CATALOGUE DISKS ... Easy to use Database Disks. They contain details of ... disks ... including Search & Print ... £2.50 ... NEW NEW

1	...	£1.00
...		

DUST COVERS

DISK LABELS

DISK BOXES

HAVE YOU HAD YOUR SAY ABOUT CU AMIGA?

Elsewhere in this issue we've printed a CU AMIGA questionnaire. This is a survey to find exactly what you think of the mag. It's not intended to be a space filler – we really want e one of you to fill it in and send it off to us. YOUR views on CU AMIGA couldn't be more im tant, as it's only through your input that we can get an idea about what you want to see i mag. You've probably noticed a lot of changes to CU AMIGA over the last few months. Ma these were a direct result of your responses to last year's survey. For instance, Art Gallery Club Call were both included because you asked for them. The same goes for Graphics Amiga Profiles and our expanded letters pages. So don't just sit there thinking it doesn't ter, grab a pen and start filling in that questionnaire now! Ta.

GAMES PROGRAMMERS AND GRAPHIC ARTISTS WANTED

A new games company urgently seeks innovative new games to publish! If you've alwa wanted to break into the software industry, we're here to help. We might be a new comp but we've got years of experience in the field of computer entertainment, so if you've got new game that you think could make waves in the computer press, why not drop us a lin show us how good you really are. We offer the best royalty payment scheme of any soft house and will endeavour to promote your game to the best of our abilities through abov below the line promotions. Send your demos to:

METRO GAMES, c/o CU AMIGA, 30-32 Farringdon Lane, Farringdon, London, EC1R 3